Automated Neuron Activity Classification in C. elegans

Arvind Seshan

Under the direction of

Dr. Vivek Venkatachalam and James Yu
Northeastern University

Research Science Institute
July 30, 2022

Abstract

Caenorhabditis elegans is commonly used to study neural activity due to 1) its well-studied genome and connectome with only 302 neurons, 2) its transparency, which makes it possible to capture detailed neural activity, and 3) its ability to self-fertilize, which allows for the maintaining of a genetically identical *C. elegans* population. Analyzing images of *C. elegans* has the exciting potential to link the activity of particular neurons to behavior or environmental stimuli. However, generating such insights has been limited by the difficulty of naming specific neurons in these images. Current approaches rely on manual classification, which is both time-consuming and error-prone. Neuron tracking systems such as ZephIR, can be leveraged to help perform labeling. However, using a tracking system in this way requires the additional step of 45 minutes of manual labeling per 11 minutes of captured imagery in order to track a specific neuron. In this study, we develop a neural-network based classifier that automatically labels sensory neurons in *C. elegans* with 91.61% accuracy. This is achieved through the use of an iterative, landmark-based neuron identification process, designed to mimic the manual annotation procedure.

Summary

The analysis of neural activity is essential to the study of brain function. Researchers use *Caenorhabditis elegans*, a transparent organism with 302 neurons, to study such neural activity. However, a major limitation in performing these studies is the ability to accurately and efficiently determine neural identities. Manual neuron classification is a difficult task due to the density of neurons within the brain and the similarity between them. In this study, we develop an automated, high-accuracy neuron classifier for the sensory neurons in *C. elegans*. The system enables a wide variety of important research including those that examine the correlations between neural activity and environmental stimuli.

1 Introduction

Caenorhabditis elegans is often used as a model organism in the field of neuroscience to study neuronal activity. It is a popular organism to study because 1) it has a well-studied genome and a connectome containing only 302 neurons and their connections, 2) it is transparent, making it easy to record activity within the body, and 3) it has the ability to self-fertilize, making it possible to maintain an essentially genetically identical population. [1, 2]. Developments in imaging techniques with cellular resolution have enabled the capture of detailed neural activity within *C. elegans* worms such as the one shown in Figure 1.

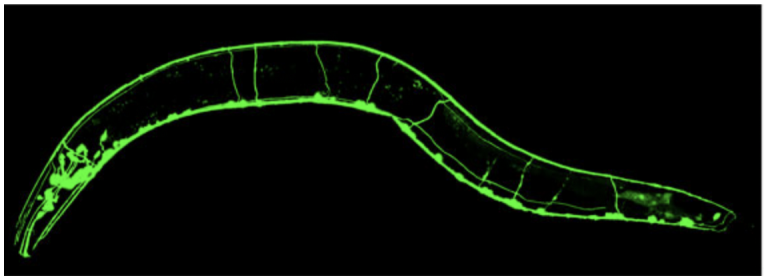


Figure 1: Fluorescent image of *C. elegans* nervous system [1].

With such detailed images, researchers can gain valuable insights into *C. elegans* neural behavior, such as the link between the activity of particular neurons and behavior or stimuli. One key step in generating these insights is identifying and labeling specific neuron activity. Unfortunately, the neuron classification and annotation process are generally done manually, a time-consuming task and error-prone task. This has proven to be a significant limiting factor in research progress in neuroscience.

Automated neuron labeling and identification are difficult because each neuron has a range of possible locations, many of which overlap with others, meaning coordinate location alone cannot be used to accurately classify each neuron. These ranges are shown by the ovals in Figure 2.

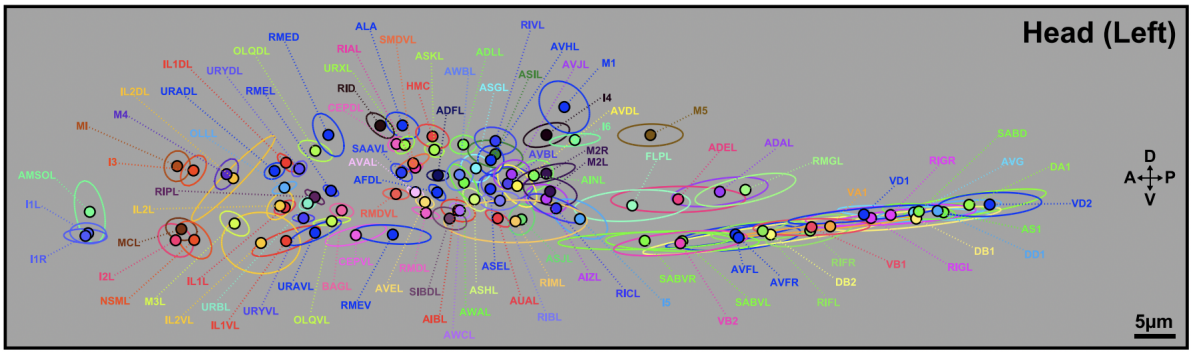


Figure 2: Range of possible locations of each neuron in *C. elegans* [3].

We propose a neural network-based classification system trained on annotated *C. elegans* videos to identify neurons within the worm, specifically the sensory neurons, which can help researchers correlate environmental factors to behavior through neuronal activity. We chose to use a neural network due to its flexibility and generalizability, given the high variation between worms indicated by Figure 2. This study aims to engineer features that aid in the differentiation of neurons in any worm, and use them to train a highly accurate neuron classification model. We train this model using three-dimensional videos of *C. elegans* being exposed to various chemicals that cause their sensory neurons to fire.

By using an iterative process that combines both location data as well as other features such as shape and fluorescent brightness, we were able to develop a neural network that correctly identified twenty-two sensory neurons with a 91.61% accuracy across six validation data sets.

The rest of the paper is organized as follows. First, we review relevant background in the field in Section 2 followed by an overview of the proposed system in Section 3, and results in Section 4. Then, a discussion of limitations and next steps is presented in Section 5, a conclusion in Section 6, a summary of practical takeaways in Section 7, and finally acknowledgments in Section 8.

2 Background

In this section, we discuss the advances in imaging techniques used in this research and current approaches for neuron identification and tracking. Below we describe existing methods for making neurons and neuron activity visible, capturing three-dimensional images, and the state-of-the-art for tracking neurons over time.

2.1 Fluorescent Neuron Marking

Two common marking techniques have been used to image neurons in *C. elegans*: GCaMP calcium indicators and NeuroPAL [3].

GCaMP calcium indicators. Current is propagated in *C. elegans* neurons via calcium spikes. As a result, measuring calcium activity can be used to detect when neurons are firing, due to the predictable manner in which calcium ions exit and enter cells. Genetically encoded calcium indicators [4] have been developed to create a fluorescent glow within the neurons, glowing more brightly to indicate the neuron has fired.

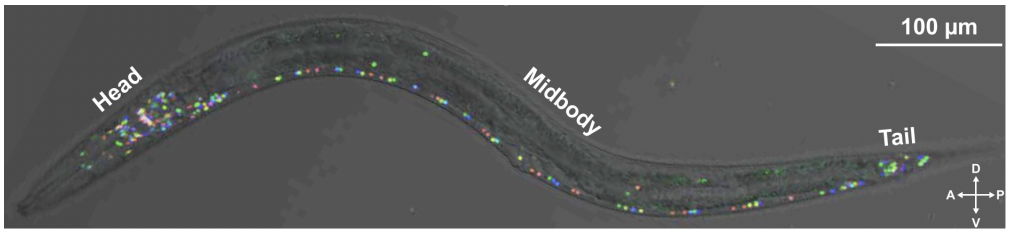


Figure 3: NeuroPAL fluorescent color map for neuron identification [3].

NeuroPAL. One drawback of calcium imaging is that neuron identification is difficult because all of the neurons are the same color and the movement of the subject can make it difficult to use location as an identifying feature. This is especially an issue since neuron identification is typically done manually by humans, a tedious process that leads to inaccuracies if not done carefully, especially due to the significant overlap in likely neuron locations.

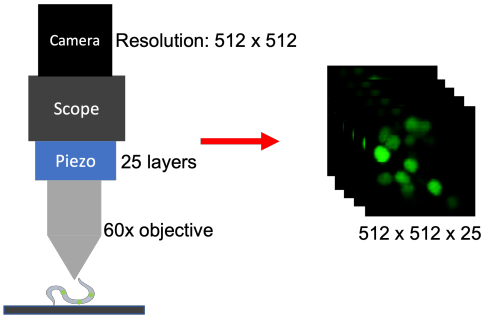
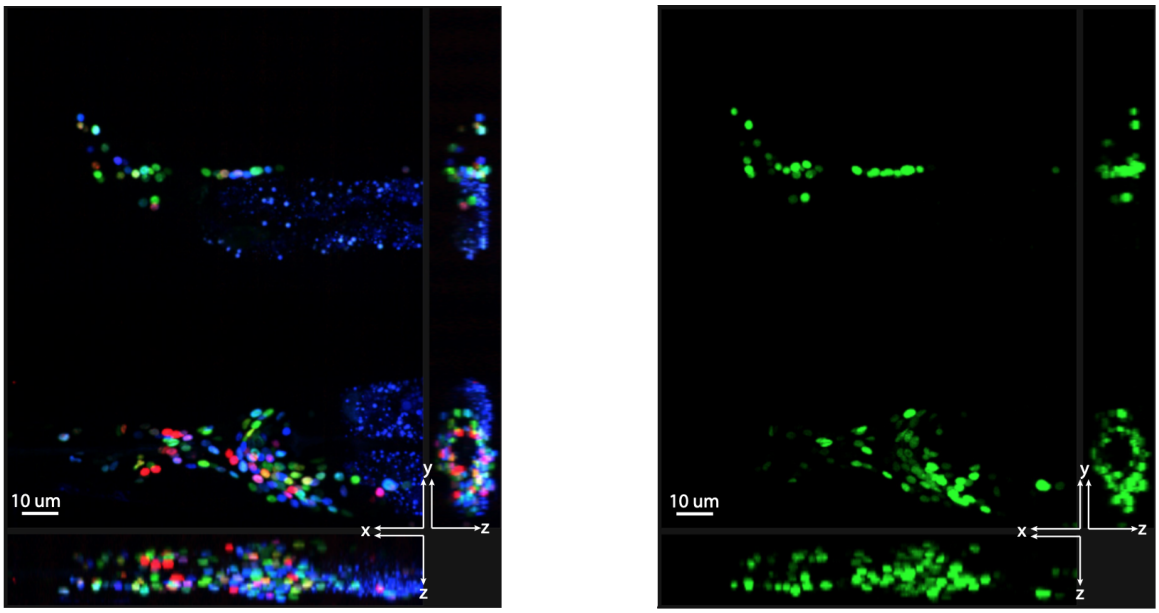


Figure 4: 3D microscope used for whole-brain neural activity imaging. Modified from [5].

A recent development, NeuroPAL [3], uses a fluorescent color map to identify all neurons by color as shown in Figure 3. This multicolor atlas is enabled by genetically modifying the worms with fluorescent reporters. However, while promising, there are several limitations with NeuroPAL, including its toxicity to *C. elegans*, which makes it difficult to perform behavioral experiments over a long duration, limiting its use to immobilized tests.

2.2 Three-Dimensional Microscopy

A multifocal microscopy imaging system, shown in Figure 4, is employed to record the three-dimensional images with single-cell resolution used in this project. The system uses the piezo attached to the 60× objective to rapidly change the focus plane. At each focal plane, corresponding to a different depth into the specimen, the camera records a 512×512 image with a 10ms exposure time. A total of 25 different focal planes are recorded, thus, generating a $512 \times 512 \times 25$ image of the specimen every 250ms or 4 volumes/s. One drawback to this approach is the lower z-axis resolution ($1.2\mu m$) compared to the x and y resolution ($0.40\mu m$). However, this drawback is not significant for our study as there is sufficient resolution to accurately capture each neuron in all three dimensions. When performing NeuroPAL imaging, a 405 nm, 488 nm and 640 nm laser light sources are used while the 488 nm laser line is used for calcium activity imaging. An example of a single plane



(a) Example NeuroPAL Image (b) Example Calcium Activity Image

Figure 5: A single plane of the images collected by the system shown in Figure 4 [5].

from a NeuroPAL and from a calcium image capture are shown in Figure 5 [6].

2.3 Neuron Tracking

Recent improvements in neuron tracking systems have enabled improvements in the efficiency of *C. elegans* analysis. One such recent tracking system is ZephIR [7]. ZephIR enables multiple object tracking within behaving animals such as the tracking of neurons within *C. elegans*. After manually annotating a set number of frames throughout videos, ZephIR can track objects such as the fluorescent signals within *C. elegans* brains over time, keeping track of which neuron is which. While this process does save time in the labeling process, there is still a significant need for human supervision in correcting incorrect frames (*e.g.*, 15% of frames in data sets in which *C. elegans* is moving). However, this development enables the creation of a large labeled data set of *C. elegans* neurons, because manually labeled data with neuron names can be passed from one frame onto all frames in a video by tracking the

neuron over time.

In this project, a ZephIR-generated data set is used to develop a neuron classification model, with the goal of eliminating the need for human involvement entirely. See Section 3.1 for more details.

3 Automated Neuron Identification System Overview

In this section, we discuss the details of the data set used in this research, the feature manipulation process, and the model architecture.

3.1 Data Set

The data set consists of sixty-five, 667.5 second long, 3D videos, each with a different *C. elegans*. The worms were from a population of hermaphrodites, due to their ability to self-fertilize. This means that each worm in the population is essentially genetically identical to the rest [2]. The calcium imaging process is used to measure the fluorescent glow in each neuron. Each video contains 2670 volume captures of $512 \times 512 \times 25$ pixels. Note that the calcium activity imaging used in the creation of this data set expresses a single color fluorescence. An example frame from these videos is shown in Figure 6 represented in two dimensions as three two-dimensional planes (xy, xz, and yz shown on the top, bottom, and right respectively). This research focuses on classifying the IFT-20 sensory neurons. The intraflagellar transport (IFT) system is responsible for the formation and function of cilia. The IFT-20 subunit of the system has been found to be responsible for the function of sensory cilia [8].

The ZephIR object tracking system [7] is used to track the neurons across time in these videos, producing coordinate locations for the neurons in each volume. The neurons are then manually labeled by their name.

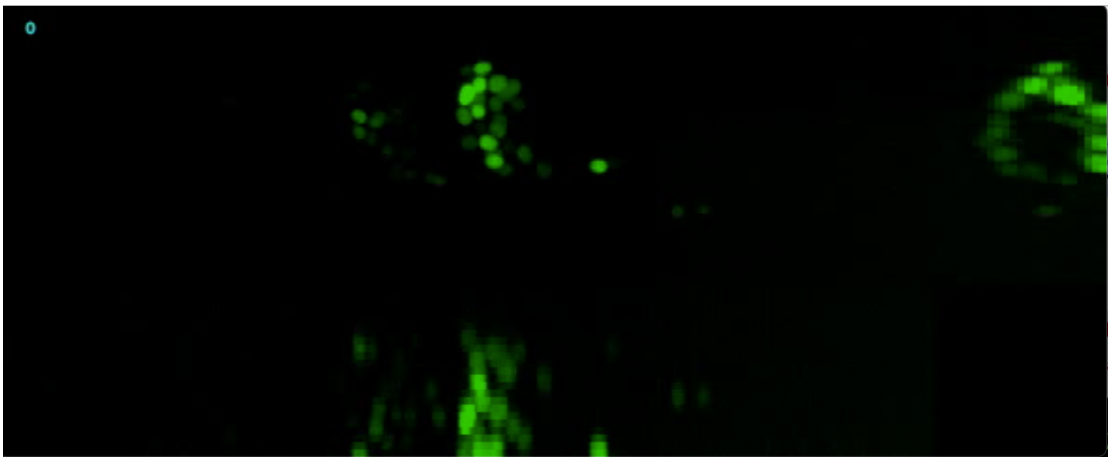


Figure 6: Example frame from videos in data set produced by calcium imaging process with single color fluorescent indicators.

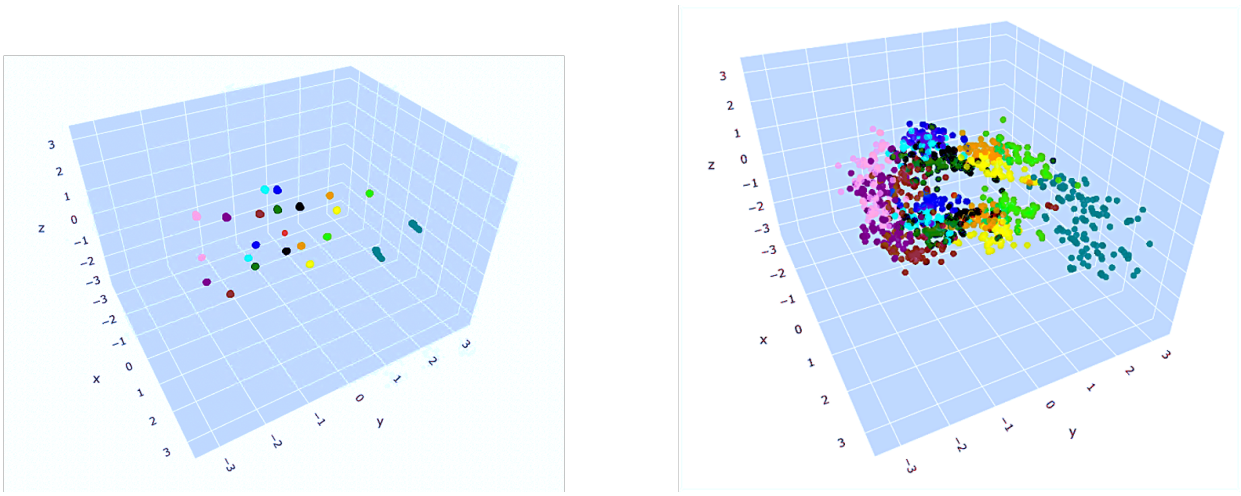
3.2 Feature Set

In order to help the machine learning model better learn from the provided data, simpler features must be extracted from the data set. Our approach to generating features that would be useful to a neuron classification model is inspired by the approach that humans use to annotate the neurons manually. A common strategy for manual labeling is to identify a few specific neurons based on a combination of location, relative isolation from other neurons, brightness and shape. Once these initial few neurons are labeled, additional neurons are labeled using their location relative to the identified ones. Based on this, we chose to use five groups of features: centroid-relative coordinates, local structure, global position, shape, and fluorescent brightness. Each of these features is described in detail below.

3.2.1 Centroid-Relative Coordinates

Neuron position in 3D space is computed using the ZephIR object tracking system, which outputs coordinates for objects it determines to be neurons in the frame.

To normalize the 3D coordinate locations of each neuron, positions are expressed relative

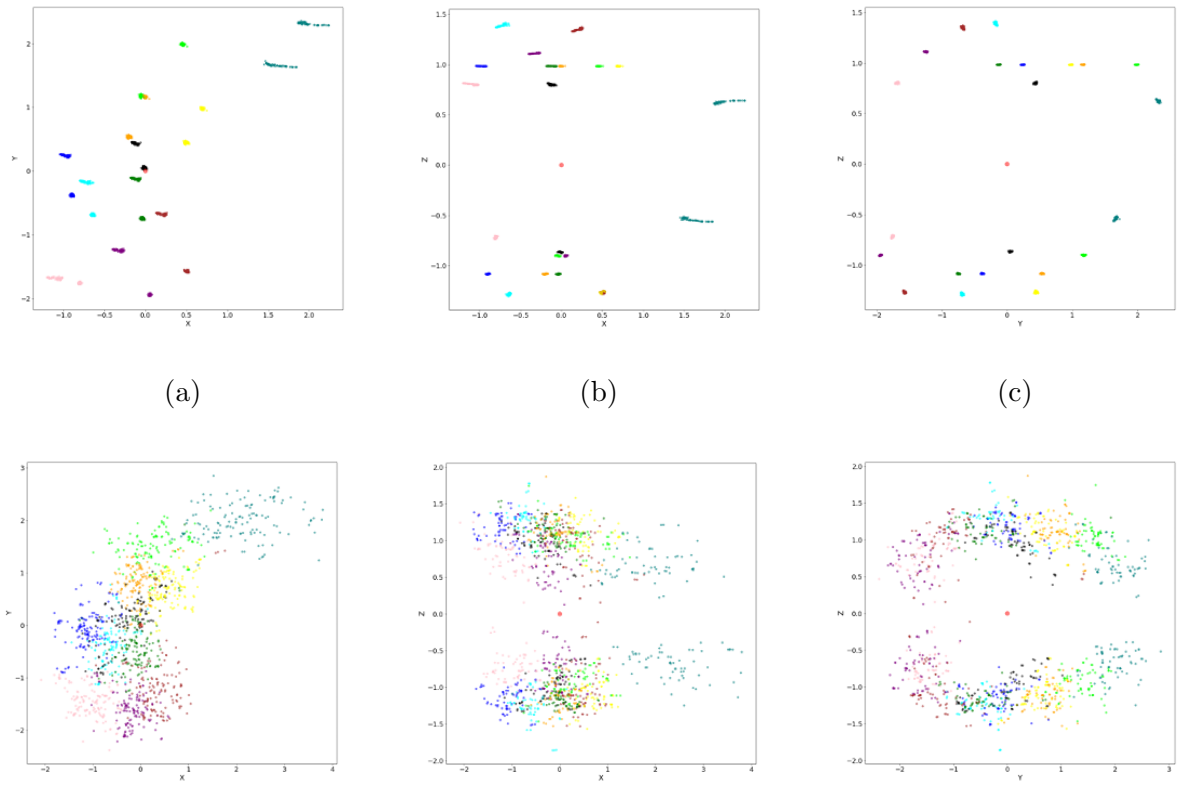


(a) One worm across 65 frames

(b) One frame from 65 worms

Figure 7: The coordinate distribution of each neuron across frames and worms respectively. Left and right corresponding neurons are colored the same.

to the centroid of the neurons. Then, each component of the coordinates is divided by the average distance away from the centroid along each axis respectively. Since an average is taken, this process is robust even when some neurons are missing in particular frames. Figure 7 shows three-dimensional plots of the coordinate distribution color-coded by neuron. Note that the *C. elegans* brains have corresponding left and right brain neurons and that these neurons are colored the same in the figure. Figure 7a shows that there are distinct clusters of neurons even across 65 randomly selected frames where the worm stretches and moves slightly. This proves the coordinate normalization system is robust and coordinates do not change as a worm moves. Figure 7b represents the coordinate distribution for a randomly selected frame from 65 different worms. Figure 8 shows the 2D projections of this same data. The standard deviation of this position information is 0.020 for 65 frames of the single worm and 0.238 for the single frame from 65 different worms. There is a much larger variation in position for each neuron and larger overlap between neuron locations across the different worms, indicating the need for other features beyond centroid-relative coordinate location when identifying a large number of neurons.



(a) (b) (c)

(d) (e) (f)

Figure 8: 2D projections of the data shown in Figure 7. (a-c) show projections to xy, xz, and yz planes, respectively, for 65 frames of a single worm. (d-f) show projections to xy, xz, and yz planes, respectively, for a single frame of 65 different worms.

Note that in addition to being a feature given to the neural network as an input, the calculation of local structure and global position below use these centroid-relative coordinates to ensure calculations such as distances are normalized.

3.2.2 Local Structure

Humans often use local structure to recognize and label similar objects. For example, it is easy to recognize and label the stars of the Big Dipper based on their relative position without knowing their overall position in the sky or their relative brightness. For each neuron, the distances to its three closest neighboring neurons in three-dimensional space are added

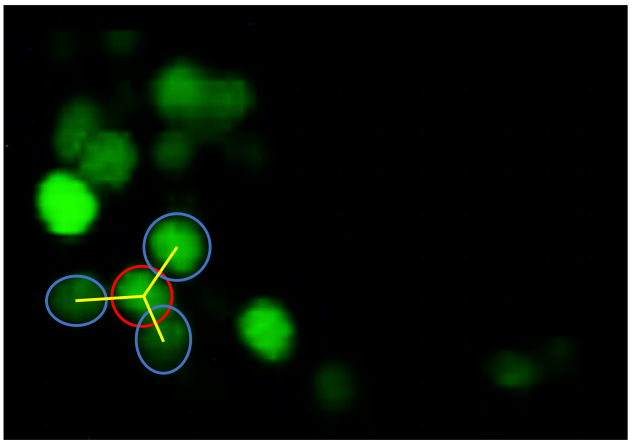


Figure 9: Closest three neurons to neuron marked in red are marked in blue. Yellow distances represent the local structure feature.

to the feature set to represent and convey this type of local structure information. Figure 9 shows an example of how this local structure feature is measured.

3.2.3 Global Position

The location of a neuron within some global context such as the worm's body or worm's overall nervous system can provide valuable information. For example, in contrast to how you recognize the Big Dipper, you can identify the North Star simply by its position in the sky, without any other information. The challenge is creating a global coordinate system to label the neuron positions in a meaningful way. Ideally, we want to give neurons position labels that minimize the variation between frames and between worms. Unfortunately, we can see from Figure 7 that using something simple, such as centroid-distance, to provide global position produces values that differ significantly across worms.

To generate a more robust coordinate system, we utilize a landmark-based approach. The position of any neuron is defined by its distance to a number of well-known landmarks. Note that the distance to a landmark localizes the neuron to the locus of a sphere around the landmark. To pinpoint a neuron in three-dimensional space, the system requires a minimum

of four landmarks, although additional landmarks can make the system more robust to measurement errors or omissions.

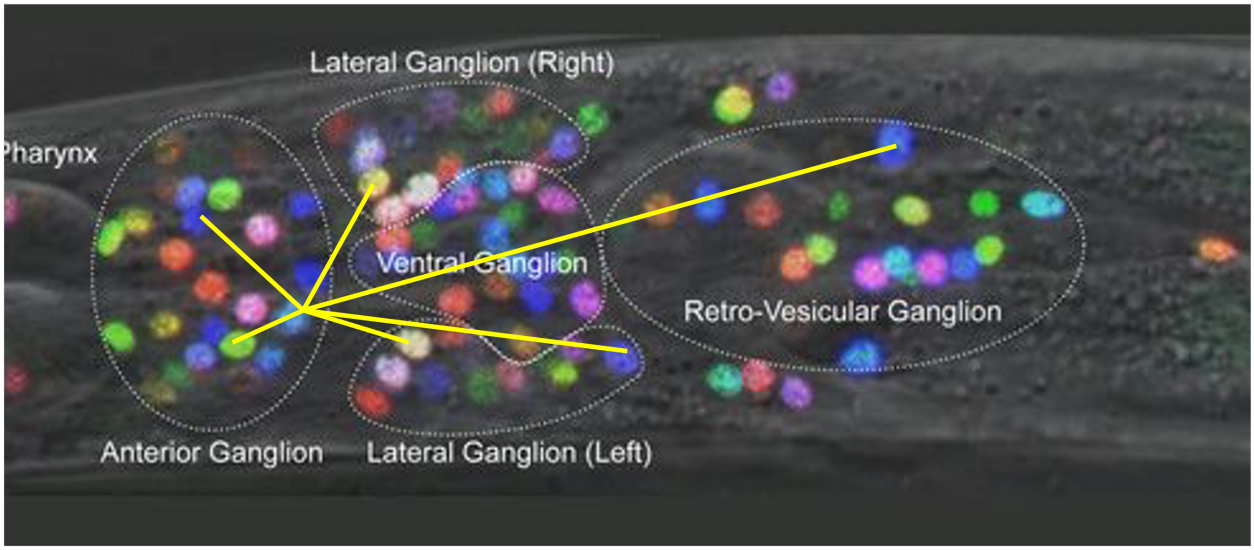


Figure 10: Landmark-based approach. Yellow distances represent the Global Position feature.

A good landmark must be easy to find in each frame in the video and should not introduce significant variability as the worm moves or deforms. We chose to use a small, easily identifiable subset of the neurons for this purpose: ASK, ADL, and ASI. Since we can use the left and right equivalent neurons in the brain, this provides six landmark neurons (ASKL, ADLL, ASIL, ASKR, ADLR, and ASIR, where L and R refer to left and right brain). These neurons move along with the rest of the *C. elegans*' brain, reducing the impact of motion or deformation as a factor. These neurons are used by manual annotators as starting reference points in neuron identification. An example of using neuron-based landmarks in this way is shown in Figure 10.

A separate neural network was designed to bootstrap this process by recognizing and labeling a small number of neurons. This special neural network had all the same inputs as the general neural network other than the global position.

Training and running this simpler network identified six neurons that could be labeled

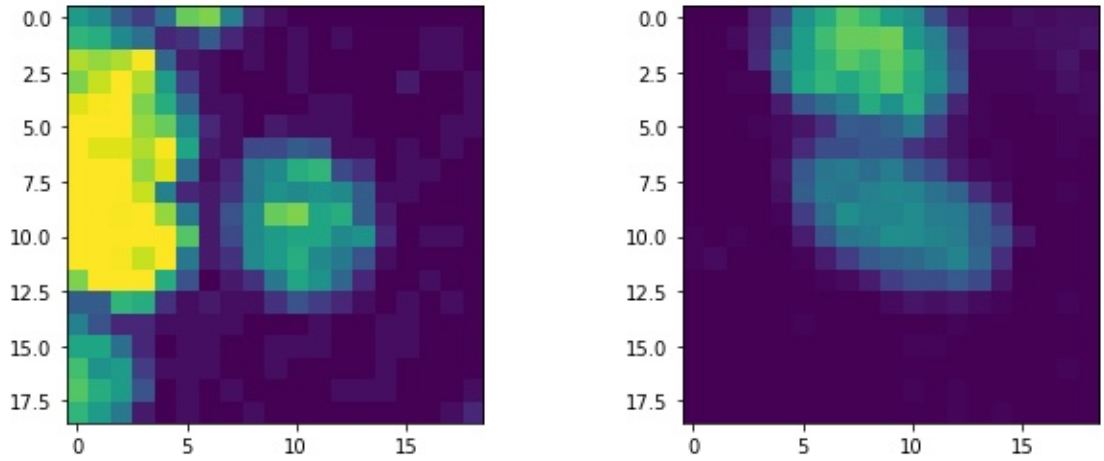


Figure 11: 2D representations of volume crops of the ASEL (left) and AWCL (right) produced by the calcium imaging process. Note the more oval shape of AWCL.

with high accuracy (98%) and could serve as landmarks. See Section 3.3 for details on the architecture of this landmark neuron classifier and Section 4.2 for evaluation of the accuracy. With the identified landmarks, the location of each neuron is expressed as a vector of six values (distances to each landmark) that are passed to the neural network as features.

3.2.4 Neuron Shape and Fluorescent Brightness

The shape of the neurons as represented in the calcium imaging process differ in significant ways. For example, the AWCL neuron in Figure 11 has a more oval shape that the ASEL. To create a feature that represents the shape of the neuron, we extract the relevant pixels from the three-dimensional images of the *C. elegans* brain. These images are cropped to capture each individual neuron and parts of surrounding neurons. The dimensions of each volume crop is $19 \times 19 \times 5$. Notice the lower size in the z dimension due to the lower resolution when performing multifocus microscopy. Two-dimensional representations of these volume crops can be seen in Figure 11. This crop provides information about the shape of the neuron as well as some of its immediate surroundings.

The brightness values in each of these volume crops need to be normalized due to diminishing fluorescence over time caused by photo bleaching in the fluorescent proteins. To normalize the brightness, the system divides by the average brightness across the neurons in a frame.

A numeric score for the brightness of the neurons is also extracted from the fluorescent values in the volume crops. The fluorescence is averaged across the center $6 \times 6 \times 2$ pixels of the $19 \times 19 \times 5$ crop collected as part of the neuron shape.

3.3 Model Architecture

The neuron classification model consists of a combination of convolutional and fully connected layers to account for the different types of model inputs. A diagram of the neural network architecture used is shown if Figure 12.

Note that the architecture contains two instances of very similarly structured neural networks [9]. The top half of the diagram shows the neural network instance used to identify the landmark locations. These landmark locations are then used to compute the distances associated with our global position feature input. These distances are passed as input to the neural network in the lower half of the diagram. Below, we describe the common components of the top and bottom neural network structure.

The collection of convolutional layers of the neural network, labeled convolution1 and convolution2 in Figure 12, receives the $5 \times 19 \times 19$ volume crops and passes them through a series of 3D convolution and batch normalization layers, reducing the dimension to an output size of $5 \times 7 \times 7$. Convolutional layers are commonly used in image processing tasks [10, 11] since they capture the spatial relationship between the pixels. In contrast, processing pixels using a fully connected layer would ignore this relationship, making each output have a potential relation to every input. This makes it more difficult to train and reduces performance. The basic step in a convolutional layer is to treat the input as a matrix and convolve

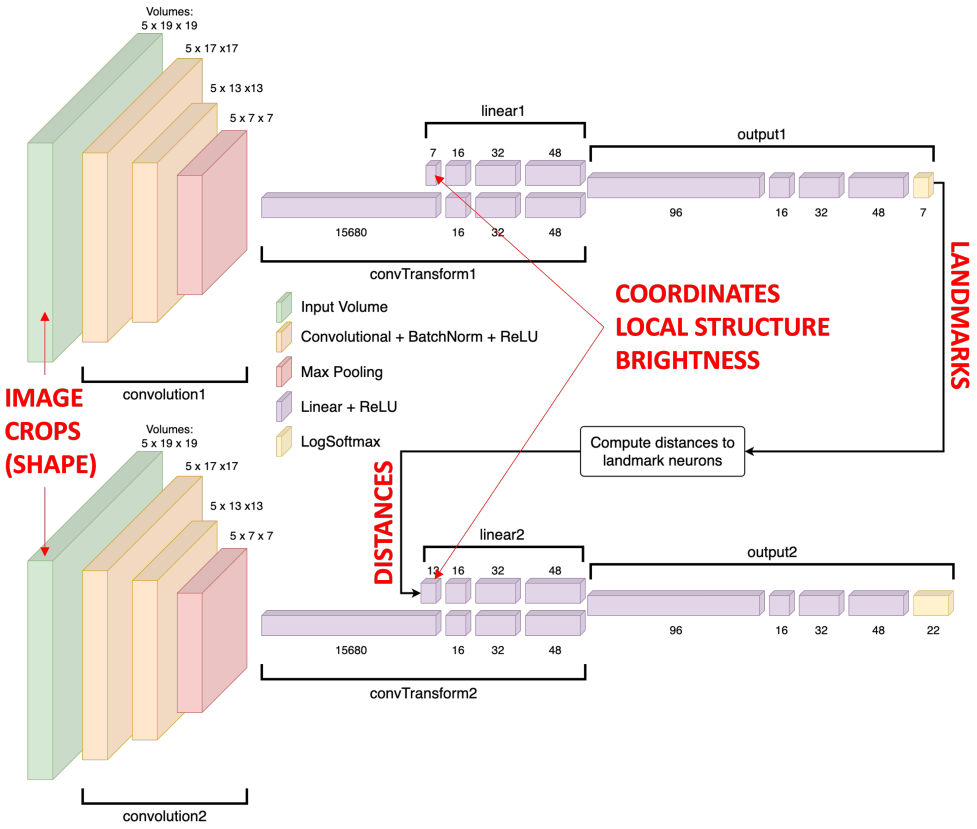


Figure 12: Neural network architecture used to label neurons in *C. elegans*.

a second smaller matrix, called the kernel, across the input, multiplying the two matrices at each step. This reduces the input into a form that summarizes key features and is easier to process. Convolutional layers are typically used along with pooling layers [12]. Pooling layers perform a similar convolution; however, extracting the maximum or average value in the smaller matrix's region at each step rather than multiplying. This essentially downsamples the information from the previous convolutional layers. Because of the reduction from the input to the output, the model is forced to choose what data is passed forward, essentially making it learn to only pass the useful parts of the volumes [11, 10].

There is also a collection of fully connected layers with ReLU activations [9, 13] labeled linear1 that receives the distances to the three closest neighbors associated with the local structure feature, the average fluorescent brightness value, and the centroid-relative x, y,

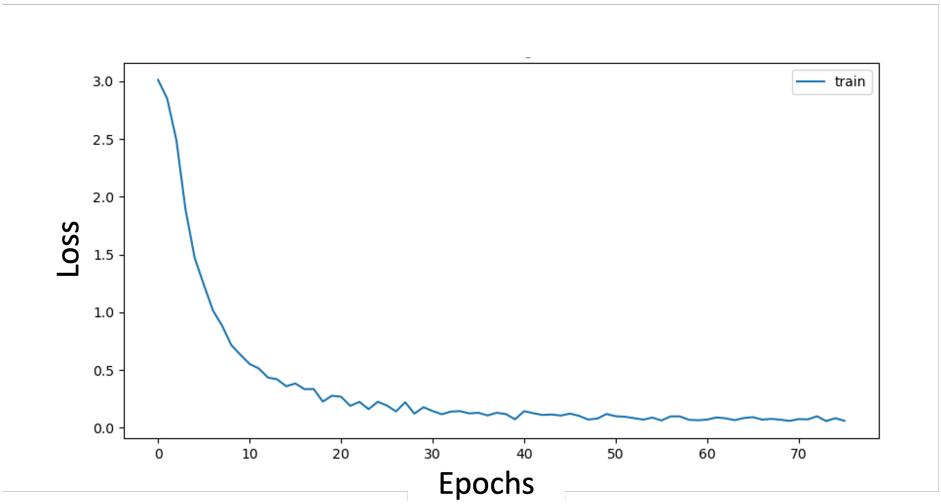
and z coordinate location in the image. The bottom instance of the fully connected layers labeled linear2 also receives the six landmark distances for the global position feature. As a result, the input shape of the top fully connected layers is 7 and the bottom is 13. This section creates a simple sequential model with a series of fully connected layers. Its output size is 48.

The output of the linear1 and linear2 sections are much smaller than that of their corresponding convolutional layers, potentially biasing the weights towards the volume data. Therefore, a set of fully connected layers labeled convTransform1 and convTransform2 are used to reduce the size of the convolutional layer output from $5 \times 7 \times 7$ to 48 to match the output of the linear1 and linear2 sections.

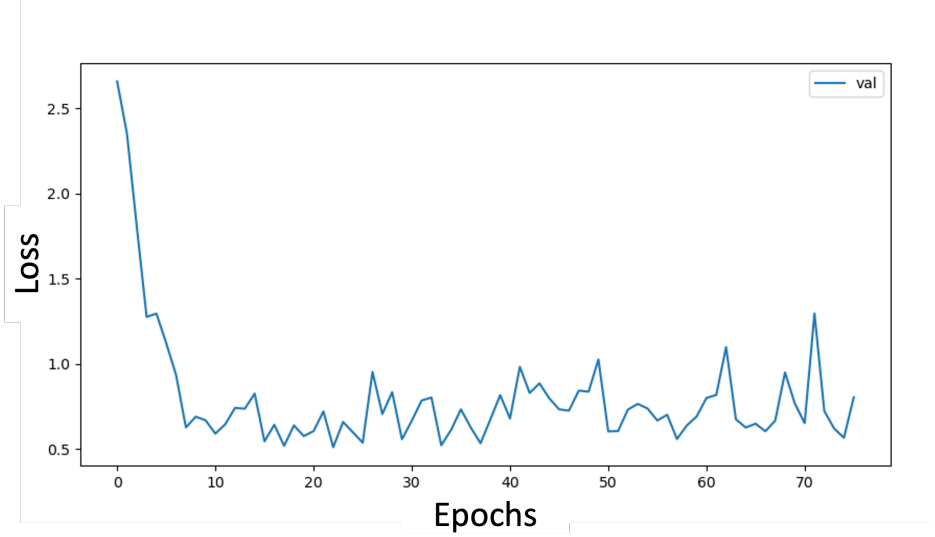
Finally, the output of the linear1 and convTransform1 are combined in as set of fully connected layers labeled output1. The final layer is a log softmax layer that produces a log probability distribution across the possible outputs. Log softmax is used as the output layer activation function because the probability distribution allows us to determine a confidence score for the predicted label. In addition, the negative log-likelihood loss function is being employed for model training. The output1 section has 7 possible neurons in the final layer, representing the 6 landmark neurons and a "not a landmark" output. The linear2 and convTransform2 outputs are processed similarly. However, the final output of the output2 stage has 22 possible neurons representing the set of IFT-20 neurons that we currently identify [9].

4 Results

To describe the performance of our neural network, we first describe the details and performance of the training process (Section 4.1). We then discuss the accuracy of the resulting model on data set aside for model validation (Section 4.2).



(a) Training loss



(b) Validation loss

Figure 13: The change in the training and validation loss during the 75 epoch training process.

4.1 Model Training

There are multiple parameters that must be configured as part of the training process. First, we use the negative log-likelihood loss (NLLLoss) function for training since it is a good function for training models that are meant to classify multiple distinct classes and was empirically determined to perform best. Loss essentially describes how far predictions

are from the true label. Second, we use the ADADELTA optimizer function [14] for training. ADADELTA is a stochastic gradient descent method that has adaptive learning rates. The goal of training is to minimize the error, which is done using backpropagation [15], which calculates the error gradient based on model parameters. Essentially, this process provides adjustments to model parameters based on what is determined to minimize loss. Finally, to prevent overfitting, we employ early stopping using the metric of validation loss and a model weight decay of 0.001. Overfitting is when a model fits too closely to training data and thus does not generalize well and performs poorly on testing data. Early stopping prevents this by stopping the training process when the validation loss increases. Weight decay keeps weights from getting too large by penalizing complexity, essentially setting a limit to how complex the model can become.

Of the 65 data sets/*C. elegans*, 59 are chosen as training data and 6 for validation. As each batch is iterated over for a maximum of 256 epochs, 10 frames are selected from each. This is because frames that are closer together tend to be more similar as worms have less time to move

Figure 13 shows the results of the training process. Notice how the early stopping system ended training early at 75 epochs when the validation loss began to rise. The training and validation losses were at 0.001 and 1.2 respectively when training ended.

4.2 Model Validation

When testing the model, we make predictions on the validation data sets by frame. This allows us to consider the fact that each neuron only occurs once in each frame, meaning that once the label has been assigned to a neuron in a frame, the possibility can be eliminated from all remaining neurons. Mimicking the iterative assignment process during the manual annotation process, the system proceeds from the highest to lowest confidence of predictions, eliminating each label as it is assigned. This ensures that each neuron will have its own unique

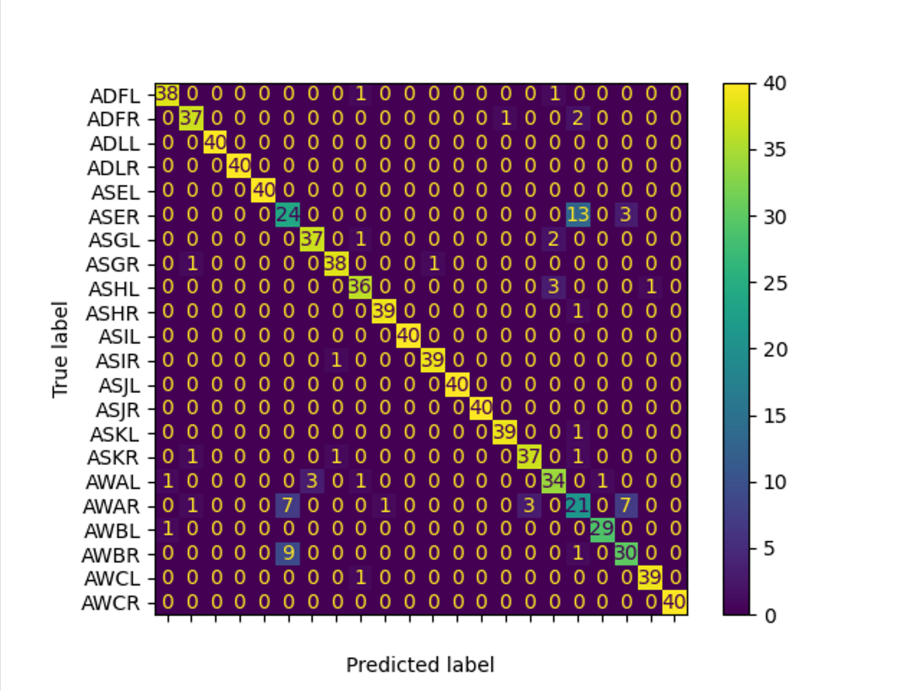


Figure 14: Confusion matrix of predictions compared to the truth when predicting on the validation data.

label within a frame. Ensuring each prediction is unique within a frame results in a validation accuracy of 91.61% when comparing predicted neuron labels to true labels using ten frames from each of six validation data sets.

Figure 14 is a confusion matrix summarizing the performance of the system. It shows the relationshsip between predictions made on the validation data set and the true labels for that data. As we expect, 91.61% of the values are along the diagonal, meaning that the true label matches the prediction.

Note three pairs of neurons show significant mislabeling with each other. AWAR is often confused with AWBR, ASER with AWAR, and ASER with AWBR. This limitation is readily explained by the anatomy of *C. elegans*, which produces significant neural overlap. For example, in Figure 15, we can see that the regions that mark the potential location of AWAR and AWBR, overlap significantly. Similarly, the other two aforementioned pairs have

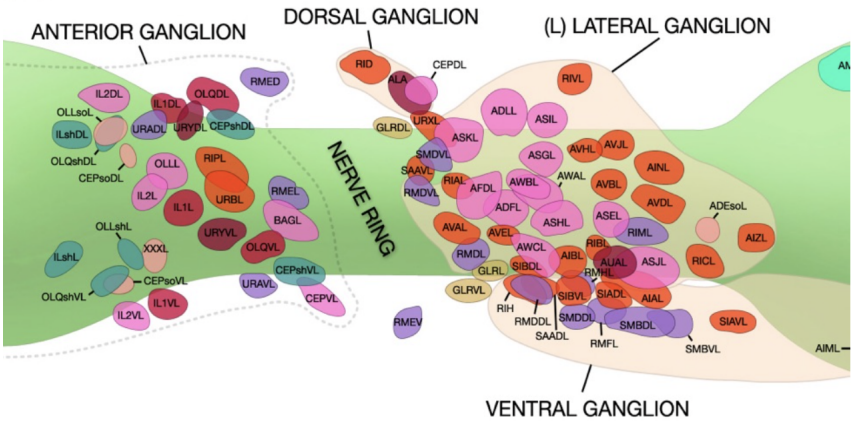


Figure 15: Proximity and overlap of neurons such as AWAL and AWBL in worm [16].

overlapping coordinate distributions. Even when manually labeling such neurons, confusion between these neurons is common.

5 Discussion and Next Steps

While the above results show that the system performs extremely well at labeling neurons, there are a few potential areas for improvement. We describe three potential enhancements to the system below.

The next step is to make this system scalable and develop a pan-neuronal classification system. The detailed feature selection process in this research likely will generalize well to the entire *C. elegans* brain, serving as a complete replacement for the NeuroPAL [3] color atlas.

As noted previously, overlapping neurons is a concern in *C. elegans* whether manually or automatically labeling them. Since neuron location was found to be the most influential feature in determining neuronal identity, the model faced difficulty in discriminating between neurons such AWAR and AWBR, whose coordinate distributions have significant overlap. This drawback could potentially be overcome by forming several binary classification mod-

els designed specifically for neuron pairs that are often confused, allowing for specialized weighting on other features such as shape and fluorescent brightness.

In addition, while the current feature set (Section 3.2) is inspired by existing approaches to manual labeling, there is one source of information that humans commonly use that our system does not leverage: cross-frame tracking. As a worm moves and reacts to the environment, different neurons are activated. In some frames of captured data, the specific neurons are much easier to accurately label. From a particular labeled image, tracking mechanisms can be leveraged to improve the labeling of both frames before and after the labeled one. One approach to creating a system of this type is to use a recurrent neural network, where the last prediction of neuron locations is kept as hidden internal state to aid in generating the subsequent predictions. To improve the labeling of earlier frames, the video can be processed in both forward and reverse since tracking mechanisms across frames are likely to be robust to the direction of playback. In essence, this approach combines the strength of the system described in this paper with previous efforts such as ZephIR [7].

Finally, we hope to try to interpret the result of applying machine learning to neuron identification to see if there is any intuition that can be used to aid other manual and automated approaches to neuron identification. This may require using alternate machine learning approaches that lend themselves better to human understanding.

6 Conclusion

This study shows that it is possible to develop an accurate and automated method to identify neurons in *C. elegans*. By using an iterative process that is landmark based, we are able to consistently identify neurons, far more easily than the manual methods used today. The neural network we designed is able to achieve an accuracy of 91.61% for classifying twenty-two sensory neurons in six validation data sets. This system has the potential to aid

in research studying the correlation between environmental changes, such as the introduction of a chemical substance, and *C. elegans* behavior.

This system opens up a vast range of applications in the study of both *C. elegans* as well as other organisms where manual labeling of neurons has been a significant limitation. The system we have created is also highly scalable to pan-neuronal applications. Applying the discoveries made in the feature engineering process regarding useful methods of differentiating neurons, the twenty-two neuron classifier designed in this study could be scaled to identify many more neurons in *C. elegans*.

7 Practical Takeaways

Automated neuron identification can enable a wide range of neuroscience research, resulting in a better understanding of brain activity. For example, the system could be used to identify the impact of external stimulants such as chemicals and identify which neurons are impacted and when.

The system constructed for this project can also have applicability beyond *C. elegans*. For example, it could be adapted to enable neuron identification in other organisms such as the zebrafish, which has recently become of much interest to neuroscientists. At the larval stage, these fish have 100,000 neurons and are transparent, making them ideal subjects for the study of neuronal activity [17].

Combined with improved tools for visualization the neural network in this study could have vast implications for further neuroscience research.

8 Acknowledgments

Thank you to Dr. Vivek Venkatachalam and James Yu at Northeastern for the guidance and support as we conduct this research. Thank you to Petar Gaydarov for feedback on structuring and writing research papers. In addition, thank you to RSI, CEE, and MIT for the opportunity to conduct this research. Finally, thank you to Mr. Wood and Dr. Judge, Mr. and Mrs. Rosen, Professor Liu and Ms. Zhang, and Dr. McQuade for enabling my participation at RSI this summer.

References

[1] L. R. Girard, T. J. Fiedler, T. W. Harris, F. Carvalho, I. Antoshechkin, M. Han, P. W. Sternberg, L. D. Stein, and M. Chalfie. Wormbook: the online review of caenorhabditis elegans biology. *Nucleic acids research*, 35(suppl_1):D472–D475, 2007.

[2] S. Brenner. The genetics of caenorhabditis elegans. *Genetics*, 77(1):71–94, 1974.

[3] E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G. E. Mena, A. D. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert. Neuropal: a multicolor atlas for whole-brain neuronal identification in c. elegans. *Cell*, 184(1):272–288, 2021.

[4] S. H. Chung, L. Sun, and C. V. Gabel. In vivo neuronal calcium imaging in c. elegans. *JoVE (Journal of Visualized Experiments)*, (74):e50357, 2013.

[5] M. Seyedolmohadesin. Characterizing whole-brain neuronal dynamics to natural and environmental chemosensory cues. PhD thesis proposal, Department of Physics, Northeastern University, 2022.

[6] A. Lin, D. Witvliet, L. Hernandez-Nunez, S. W. Linderman, A. D. Samuel, and V. Venkatachalam. Imaging whole-brain activity to understand behaviour. *Nature Reviews Physics*, 4(5):292–305, 2022.

[7] J. Yu, A. Nejatbakhsh, M. Torkashvand, S. Gangadharan, M. Seyedolmohadesin, J. Kim, L. Paninski, and V. Venkatachalam. Versatile multiple object tracking in sparse 2d/3d videos via diffeomorphic image registration. *bioRxiv*, 2022.

[8] A. R. De-Castro, J. Quintas-Gonçalves, T. Silva-Ribeiro, D. R. Rodrigues, M. J. De-Castro, C. M. Abreu, and T. J. Dantas. The ift20 homolog in caenorhabditis elegans is required for ciliogenesis and cilia-mediated behavior. *Micropublication Biology*, 2021, 2021.

[9] C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, 2006.

[10] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[12] J. Weng, N. Ahuja, and T. Huang. Learning recognition and segmentation of 3-d objects from 2-d images. In *1993 (4th) International Conference on Computer Vision*, pages 121–128, 1993.

[13] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[14] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[16] D. Hall. Wormatlas - a database featuring behavioral and structural anatomy of caenorhabditis elegans and other nematodes. www.wormatlas.org, 2022.

[17] A. A. Wanner and A. Vishwanathan. Methods for mapping neuronal activity to synaptic connectivity: lessons from larval zebrafish. *Frontiers in Neural Circuits*, 12:89, 2018.