SomaNet: Segmenting Interneurons in Forebrain Assembloid Microscopy

Julia Schaepe, Ashi Agrawal
jschaepe@stanford.edu, ashi@cs.stanford.edu
Department of Computer Science, Stanford University

Introduction

The Pasca Lab in Stanford's Department of Psychiatry and Behavioral Medicine recently developed a powerful human 3D brain organoid platform. Using their video data, we hope to build a platform to quantify the saltatory migration of interneurons to the cerebral cortex, but the first step requires segmentation of interneuron somas. We construct a soma segmentation dataset from interneuron migration videos, implement data augmentation, and perform transfer learning on a UNet architecture trained on non-neuron nucleus segmentation data.

- Input: 256x256 RGB images cropped from interneuron migration videos
- Output: 256x256 black and white segmentation mask

Data and Features

We train the U-Net on 670 images of non-neuron nucleus segmentation from the Kaggle 2018 Data Science Bowl [1]. For transfer learning, our hand-labeled dataset includes 32 training examples and 5 validation examples. The UNet architecture is designed to achieve high performance even with small datasets. Additionally, we augment our dataset by implementing elastic deformation (with an alpha of 34 and a sigma of 4) and mirroring. The full augmented training set consists of 96 training examples.

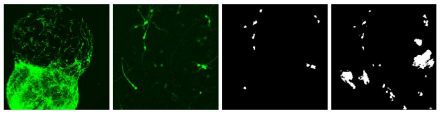

Fig. 1: Interneuron migration in forebrain assembloid (left), input image (middle left), ground truth mask (middle right), baseline predicted mask (right)

Model

We implement a UNet architecture to serve as a basis for our cell segmentation network. This architecture consists of combinations of 3x3 2D convolutions, batch normalization, ReLU layers, 2x2 max pooling layers, and transposed 2x2 2D convolutions. We utilize binary cross entropy, defined below, as our loss function.

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n) \right].$$

Fig. 2: U-Net CNN model architecture [2]

Results

We tuned the following parameters for our UNet: learning rate, batch size, and weight decay. All training was performed on 32 training examples and all testing on 5 validation examples.

model	lr	batch size	epochs	decay	layers retrained	loss	training iou	test iou
baseline	N/A	N/A	N/A	N/A	0	N/A	0.10	.07
1	1.00E-03	8	50	1.00E-04	3	0.016	0.39	0.50
2	1.00E-02	8	50	1.00E-04	3	0.017	0.17	0.25
3	1.00E-04	8	50	1.00E-04	3	0.018	0.25	0.34
4	1.00E-03	16	50	1.00E-04	3	0.027	0.22	0.32
5	1.00E-03	32	50	1.00E-04	3	0.028	0.24	0.37
6	1.00E-03	8	50	1.00E-03	3	0.028	0.22	0.34
7	1.00E-03	8	50	1.00E-05	3	0.027	0.25	0.34

Tab. 1: Model comparison for hyperparameter tuning

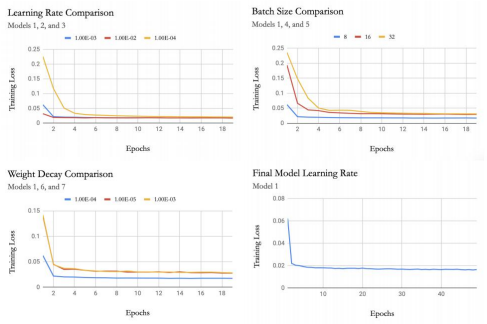

Fig. 3: Training loss: learning rate comparison (top left), batch size comparison (top right), weight decay comparison (bottom left), model 1 (bottom right)

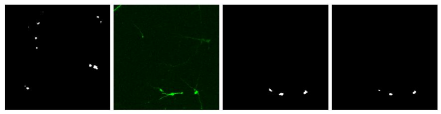

Fig. 4: Validation results with model 1: predicted mask for input image in Figure 1 (left), input image (center left), ground truth mask (center right), predicted mask (right)

Discussion

The model visually produces segmentation masks similar to the ground truth segmentations. We optimized our model for IOU and satisficed acurracy and precision of recognizing and segmenting individual somas. With our final model we achieved an average validation IOU of 50%, accuracy of 94%, and precision of 62%. Overall we are satisfied with this model as a baseline, but hope to improve these metrics to reach an average IOU of 60%. Our model is limited by the amount of data available for transfer learning due to the time-consuming nature of preparing and segmenting our own novel data. Utilizing transfer learning on a pre-trained UNet architecture proved to be successful in the segmentation of neuron somas and can provide a base for tracking interneuron movement.

Future Work

1. Video Segmentation: We would like to extend our network to be able to track the neurons in a moving video. This involves adapting the input and output of the network to take videos instead of individual frames.
2. Tracking: By segmenting these videos, neurons no longer overlap with each other as frequently and are much more clearly defined. Therefore, we hope to utilize idtracker.ai to track these videos [5].
3. Data Processing: From the tracked videos, we then would like to analyze them to quantify average velocity, direction, and saltation frequency of internueron migration.
4. Compare: We would like to compare the performance of this network to YOLO's performance for tracking [4].

Acknowledgements

We would like to extend a special thanks to the Pasca Lab, and specifically Fikri Birey and Themasap Khan, for supplying neuronal images for us to use as training data, as well as the idtracker.ai authors who kindly responded to our questions early in the learning process. [3]

References

[1] Booz Allen Hamilton. *Kaggle 2018 Data Science Bowl*. Available at `https://www.kaggle.com/c/data-science-bowl-2018/data`. URL: `https://www.kaggle.com/c/data-science-bowl-2018/data`. (accessed: 02.15.2019).

[2] Limingwu8. *U-Net Pytorch Github*. Available at `https://github.com/limingwu8/UNet-pytorch`. URL: `https://github.com/limingwu8/UNet-pytorch`. (accessed: 02.15.2019).

[3] Philipp Fischer Olaf Ronneberger and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Springer International Publishing Switzerland* (2015), pp. 234–24.

[4] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[5] Francisco Romero-Ferrero et al. "idtracker. ai: tracking all individuals in small or large collectives of unmarked animals". In: *Nature methods* (2019), p. 1.