Please do not post this final writeup to the course website.

**Neurosieve**

***Introduction***

It is common that researchers would like to trace neuron images to gain an idea and understanding of the neurons’ morphology. For example, understanding the morphology of neurons in the brain is necessary to understanding their functional role. In the past, researchers would have to trace and label each neuron by hand - a tedious and time consuming task. Recently, there have been several attempts to automate this tracing process through various software and tracing algorithms in hopes of reducing tracing time from several weeks to only a few hours. Unfortunately, due to variation in microscope quality, sample quality, and other factors, neuron images obtained using optical microscopy are often marred by background noise, reducing the performance of many automatic tracing algorithms. Neurosieve is an attempt to solve this issue by using neural networks to perform segmentation on the neuron images, removing background noise, and then running the tracing algorithms for better performance and accuracy.

***Group members and work division***

* Hasan Khan
  + Processed the data ….. (Explain more)
* Crystal Gong
  + Wrote the architecture of the program… (explain more)
* Puja Soni
  + How to trace images using Vaa3d plugin (APP2)
  + Wrote how to determine performance of Neurosieve by computing accuracy by comparing numpy arrays
* Charu Mishra
  + Determined best way to trace images using Vaa3d—SmartTracing vs APP2
  + GPU resources—tried implementing AWS P2 instance to run on
  + Helped with accuracy determination

***Milestones & Challenges***

* A description of which milestones were achieved and which challenges if any were encountered.
  + Hasan got all the data that were originally in v3dpbd and swc files and converted them into npy arrays that were actually usable by us
    - It took a really long time to figure out what format the swc files were in and the standards that the documentation online said they were in were not followed by people who submitted the data to bigneuron which cause a lot of headaches
  + Crystal found a usable architecture that was simple and especially for medical image segmentation and converted it to be usable with 3d images
    - We ended up failing to get a GPU because amazon web services ended up charging us way more money than the amazon educate gave us in money
    - So we ended up using only the 2D algorithm
  + Puja and Charu got the tracing algorithm to work and figured out how to compare accuracies
    - There was no way to get the tracing algorithm to work in batch. You have to use the program written by the owners of BigNeuron, Vaa3D, and download a plugin to use in the GUI that traces only one at a time.

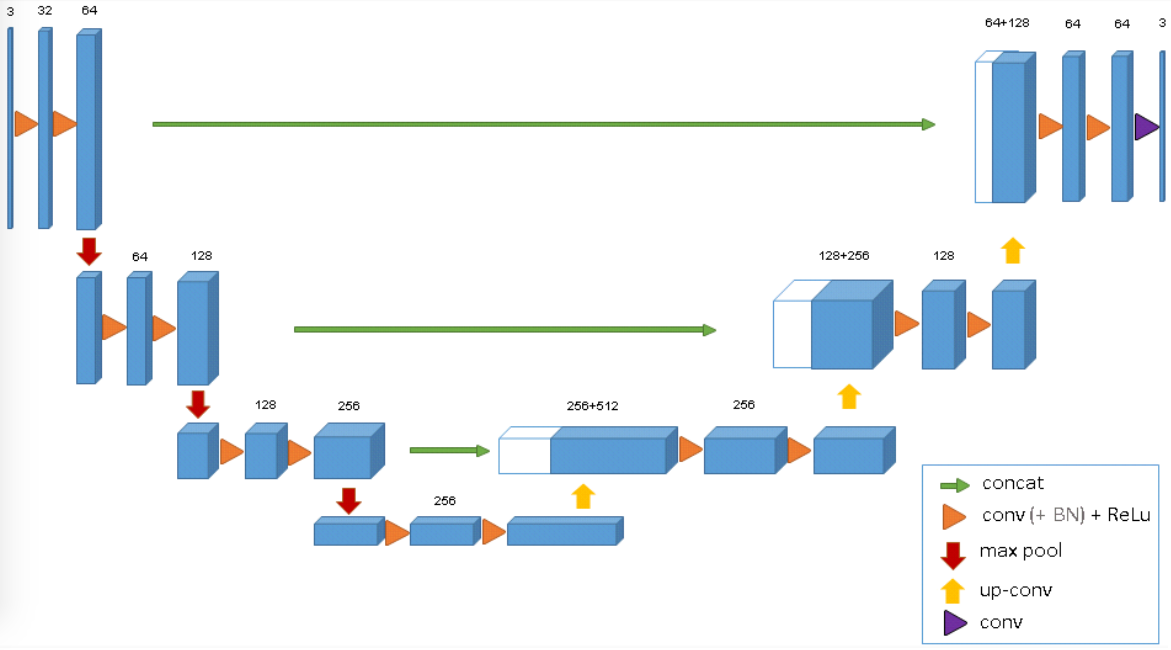
***Results***

* Figures for results, e.g. performance and confusion matrices of classifiers, images produced for graphics projects, other visualizations, etc.



* Explanations or figures for the architectures of neural networks used (if appropriate)





* Any other observations you would like to make.

***Conclusions and Future Work***

* Conclusions

For future directions, we hope to implement data augmentation such as flipping, shifting, and recoloring the neuron images. Also, it would be interesting to use different tracing algorithms other than just APP2, such as SmartTracing, to gain more insight about the performance of these tracers.

* Test on our own data

U-net 2d architecture from O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI, pp. 234–241, Springer, 2015.

U-net 3D architecture from O. Cicek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, *3D U- Net: Learning Dense Volumetric Segmentation from Sparse Annotation*, 2016, [online] Available: .