JOHNS HOPKINS
BIOMEDICAL ENGINEERING

Deep learning-based Neuron Detection in Brain CLARITY Imaging

Members: Prerna Singh

Mentor: Dr. Jeremias Sulam

Background: CLARITY imaging

Clear

Lipid-exchanged

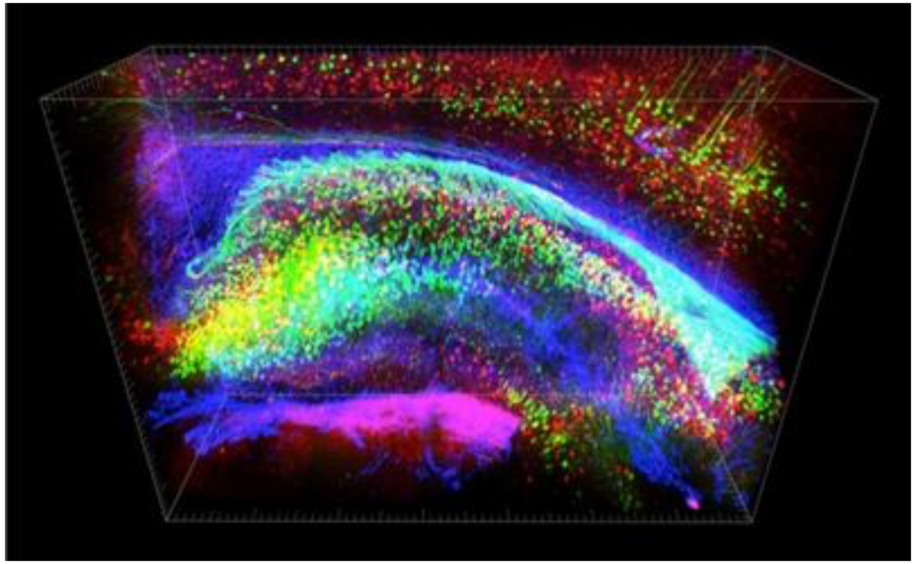
Acrylamide-hybridized

Rigid

In-situ hybridization-compatible

Tissue

hYdrogel



https://en.wikipedia.org/wiki/CLARITY

Significance of CLARITY for Brain Imaging

- Connectome project
  - Local circuit wiring
  - Relationships between neural cells★
- Neurological diseases
  - 3D view of brain structures

http://www.humanconnectomeproject.org/

Goal

- Develop a robust 3D CNN that can predict, with improved accuracy when compared to other models, how many fluorescent neurons are present within a section of a brain imaged with CLARITY

  - Previous models: template matching, blob detection
  - Maximum accuracy achieved was ~ 59% with these techniques

Deliverables

Minimum:

- Trained 3D CNN model (pytorch) for neuron detection ✓
- Report in Jupiter notebook – explains training of CNN and predictions ✓

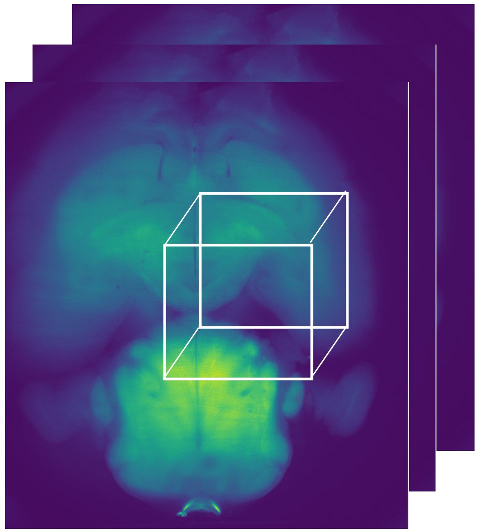
Expected:

- Robust and validated trained model (pytorch) for neuron detection ✓ Still working on it

Maximum:

- Academic paper describing packaged model, training, validation
- Surpass baseline accuracy of previous models by a significant margin ✓

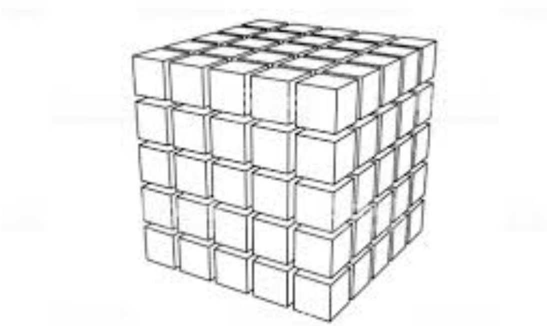
Technical Approach- CNN Architecture



Input: any 3 dimensional subset of a CLARITY Brain scan

Trained 3D Convolutional Neural Network

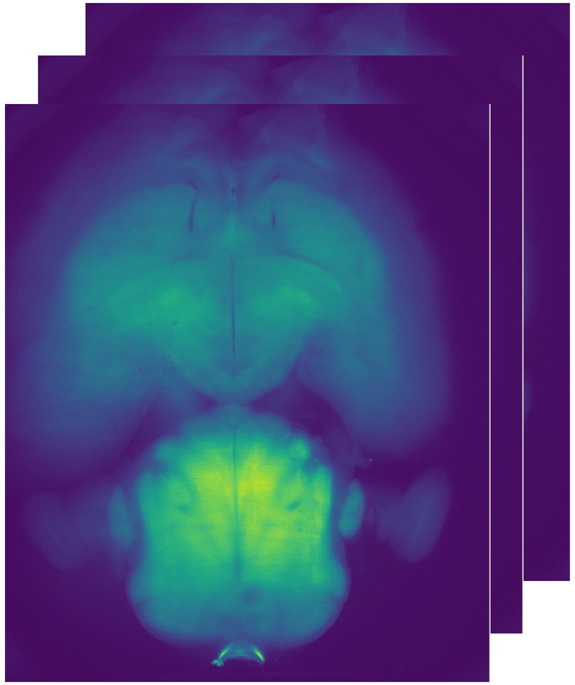
Layer 1: 3D Convolutional
Layer 2: Maxpool
Layer 3: 3D Convolutional
Layer 4: 3D Convolutional
Layer 5: Fully Connected
Layer 6: Fully Connected



Output: score of 0-1 of neuron probability for each 12x12x12 voxel in input

https://www.istockphoto.com/vector/abstract-3d-cube-from-cubes-vector-outline-illustration-gm1034794628-277033503

Technical Approach – First training and validation of 3-dimensional CNN training on reduced dataset ✓



One 3-D CLARITY Brain Imaging Data annotated for Neurons

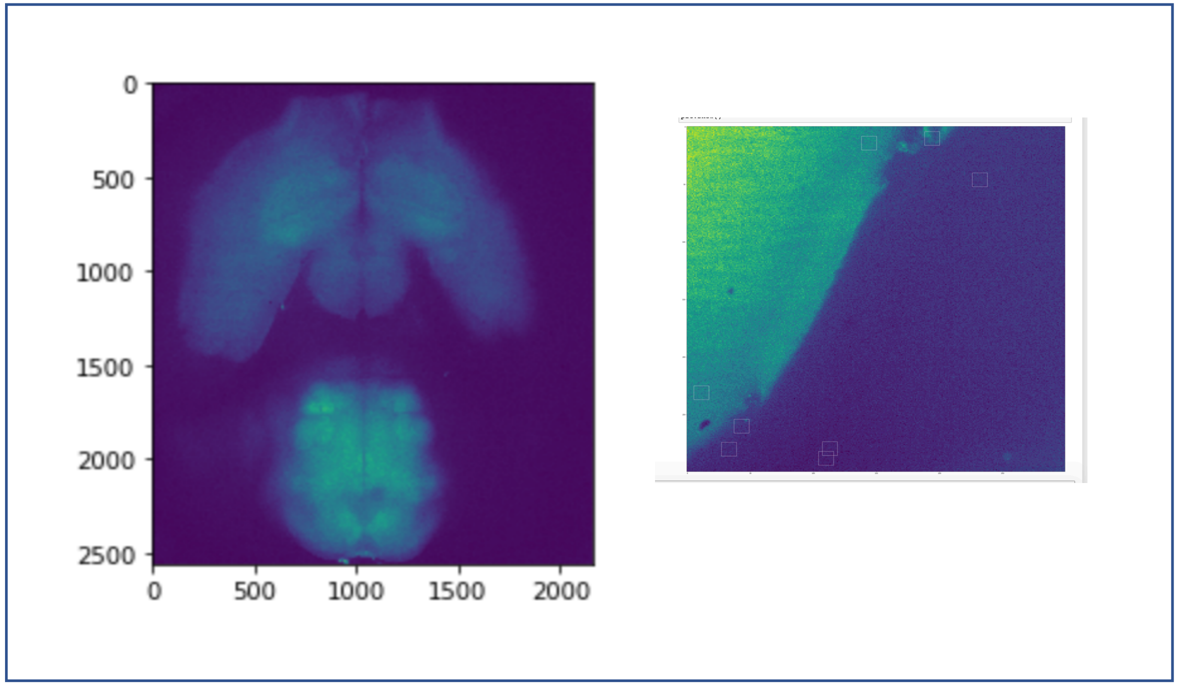
Neuron 12X12X12 voxels and non-neuron 12x12x12 voxels from these images are used for training

3D Convolutional Neural Network

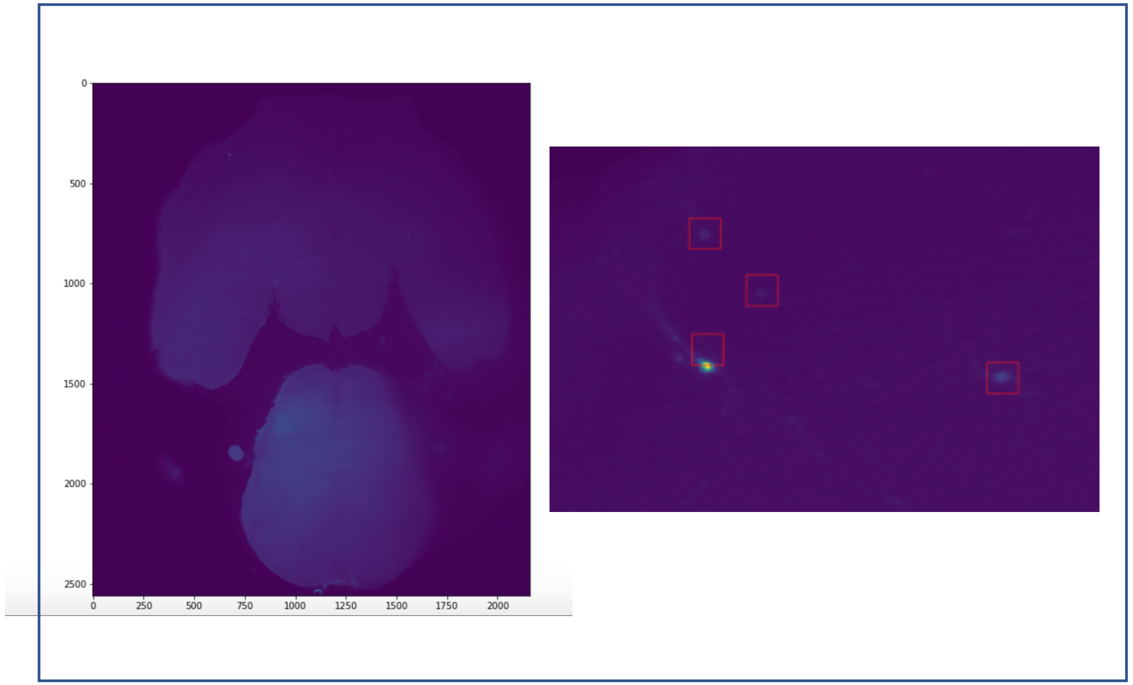
Model resulted in 99% accuracy

Problem 1: Variation in Brain Intensity

Before



After

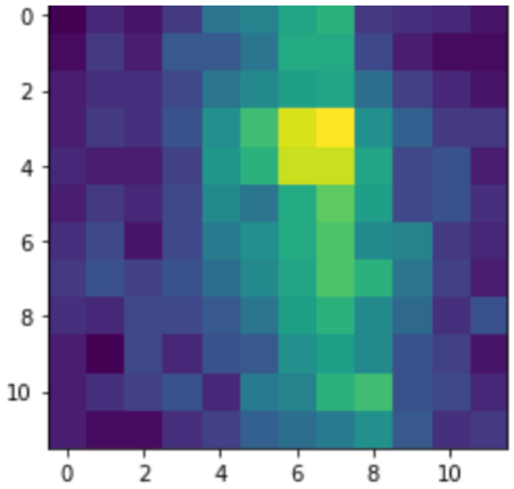


Solution: Brain illumination correction filter applied to the data

Problem 2: How to Extract Training Data From CLARITY Scans

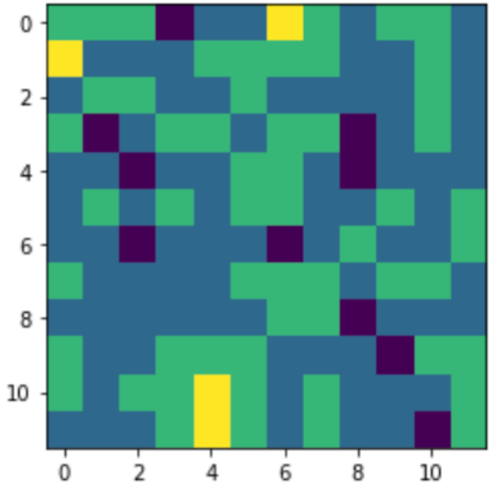
- Non-cell data should be representative of brain tissue , not background

Cell Data- good



Non-Cell Data- bad

Non-Cell Data- better



- Solution: Pick non-cell regions within 6-20 pixels of an annotated cells

Technical Approach- Large Scale Training on Complete Dataset & CNN Validation

- Data includes 8 brains
- 2376 Cell Anotations
- 23760 Non-Cell Regions Extracted

- Coarse validation: report metrics over a few random splits of data (5 train, 3 test) ✓
- Full Leave-one-out (LOO) Cross validation: report cross-validated metrics averaged over 6 folds. In each fold, all but 2 brains will be used for training, and the remaining 2 for testing ★

Metrics Reported

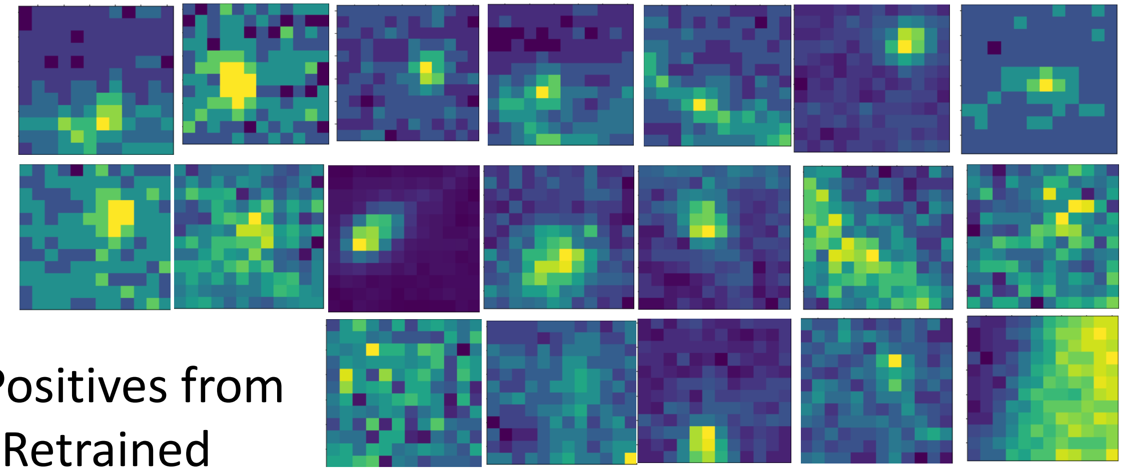
- F1 score: harmonic mean of precision and recall
- Accuracy = (TP+TN)/(TP+TN+FP+FN)
- Precision= TP/(TP+FP)
- Recall= TP/(TP+FN)
- False Positives: non-cell regions classified as cell regions
- False Negatives: cell regions classified as non-cell regions

*Metrics reported at threshold score of 0.5

Coarse Validation Results

Split	F1 score	Accuracy	Precision	Recall	FP	FN	AUC ROC	AUC PR	Adjusted FP
1	0.7612	0.9647	0.9867	0.6203	8	77	0.924	0.817	4
1 retrained	0.7784	0.9664	0.9733	0.6485	17	49	0.935	0.840	9
2	0.7730	0.9657	0.9701	0.6425	12	61	0.920	0.805	8
2 retrained	0.7251	0.9596	0.9493	0.5865	19	76	0.909	0.775	5

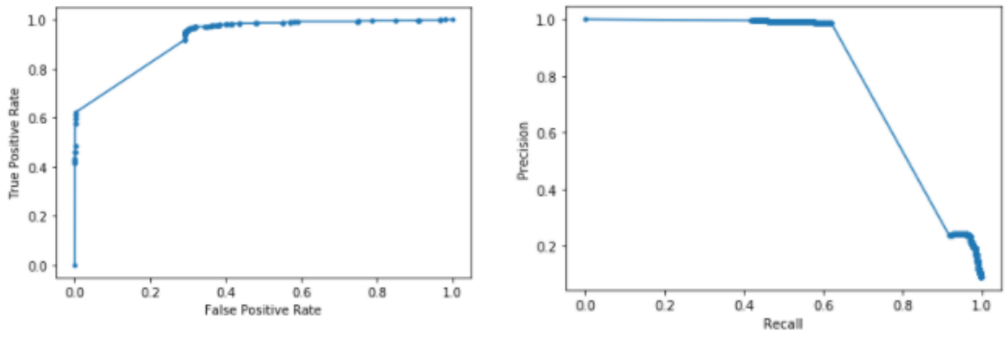
*Retrained include True False Positives and all False Negatives



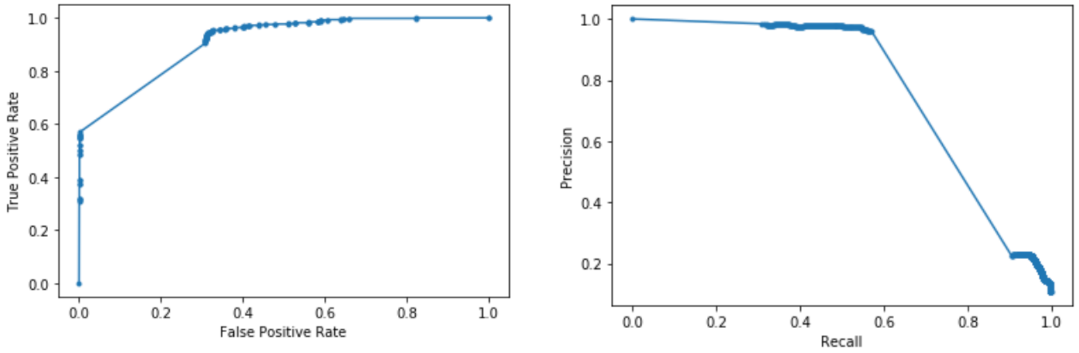
Ex) False Positives from Split 2 Retrained

Coarse Validation Results

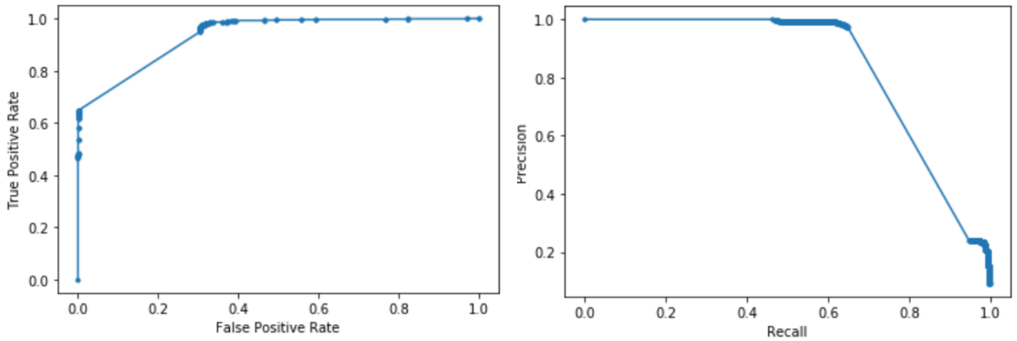
Split 1



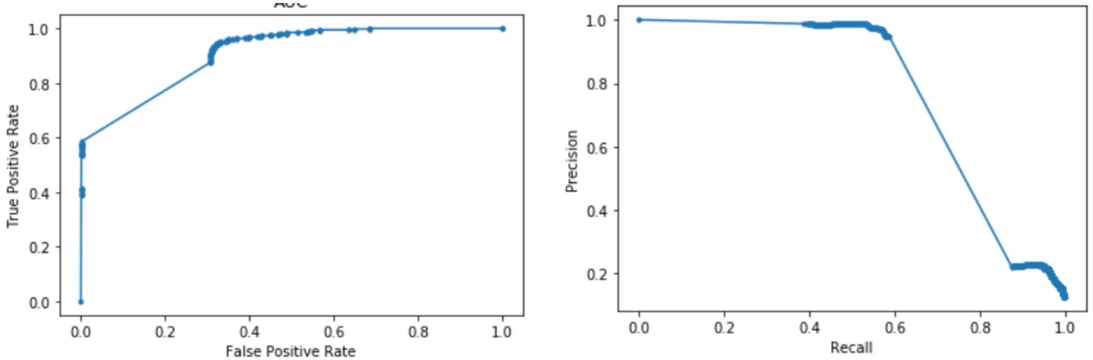
Split 2



Split 1 Retrained



Split 2 Retrained



Work Remaining

- Cross-Validation
- Academic Paper

Management

- Meetings with Dr. Sulam as necessary
- Code stored on github and MARCC
- Code backed up to github every week and more often with substantial improvements
- Communication through Slack/email

Lessons Learned

- Budget more time deemed necessary for computational difficulties (MARCC)
- Quality of training data is very important