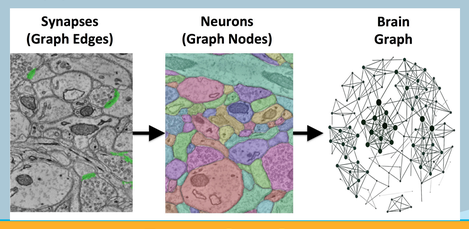
Analysis of the Performance of Vesicle Detection Systems

Gregory Levine, Matt Saltzman, Alex Sharata, Greg Kiar, William Gray Roncal

Department of Computer Science, Johns Hopkins University, Baltimore, MD

Abstract and Motivation

- Researchers are only just beginning to explore the neuro Electron Microscopy (EM) data, and there are many things about the structure of the brain to be learned from it.
- Identifying synapses, the key to analysing EM data, is hard but identifying neurotransmitter containing vesicles can be used as a contextual clue to identify synapses.
- A method for vesicle detection has already been implemented in Matlab [1], but there is no similar function in Python.
- We will test the precision/recall metric of this algorithm while also developing a parallel Python based pipeline.



Introduction

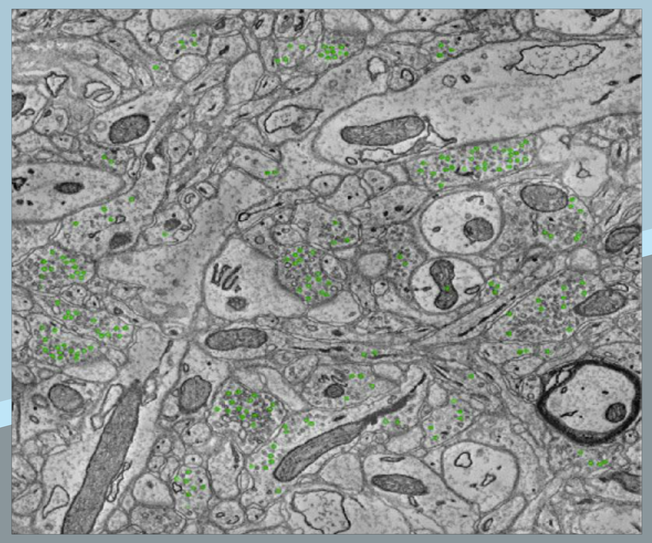
Identification of synapses is the key to the development of complex neural graphs and the burgeoning field of connectomics. Using contextual clues is very useful to any synapse detection algorithm and because synapses in the brain have vesicles adjacent to them, vesicle detection software can have great utility in a synapse detection pipeline.

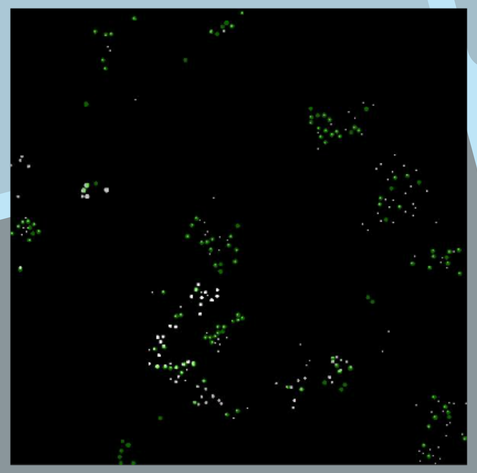
Before our project, this software had been developed in MATLAB however it was not tested for precision/recall and did not exist in any other languages. Therefore, for our project we decided to take a two pronged approach in testing the existing data and attempting to establish a new vesicle detection pipeline in Python.

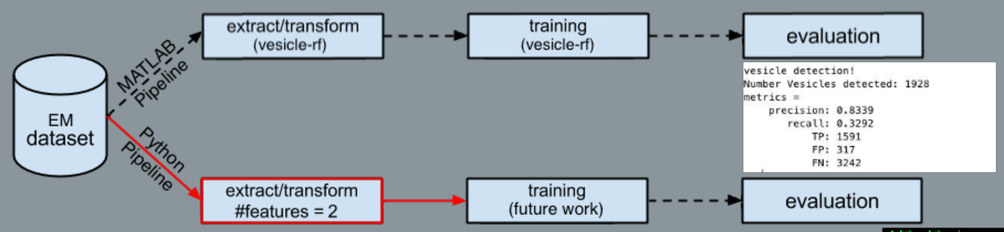
Two key steps are needed in the an algorithm such as ours based in computer vision. There is a feature extraction step in which potential objects you are searching for are labelled and identified. After that feature extraction "supervised learning" must be performed to train the algorithm to identify objects with precision and accuracy. In essence, multiple data sets with vesicles already identified must be fed into the algorithm and the computer will teach itself to be more accurate based on what it got right and what it got wrong.

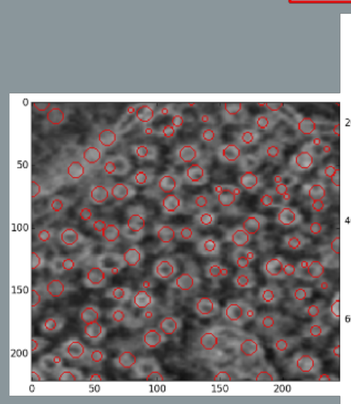
Design, Method, and Results

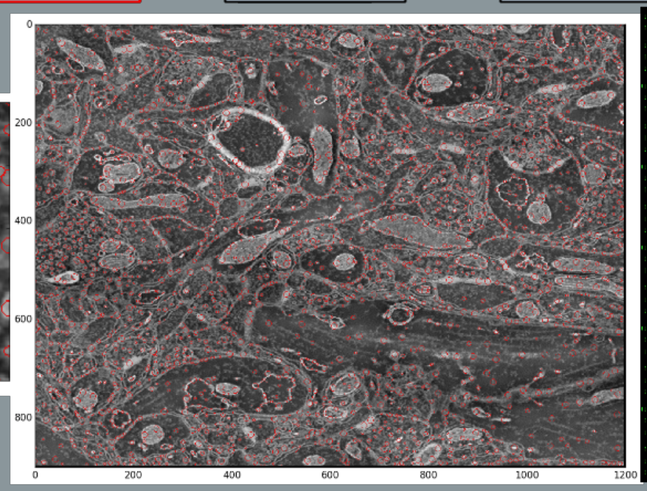
- First, we analyzed the performance of the existing vesicle detection algorithm in Matlab:
  1. Ran the detector on EM data from neurodata.io, using it to generate vesicle annotations
  2. Compared the generated vesicle annotations with annotations made by hand by experts
  3. Determined True Positives, False Positives, and False Negatives, and computed Precision/Recall

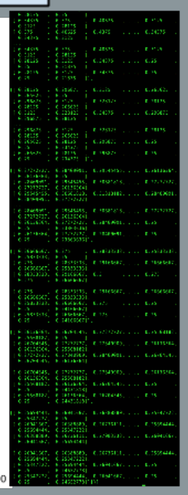




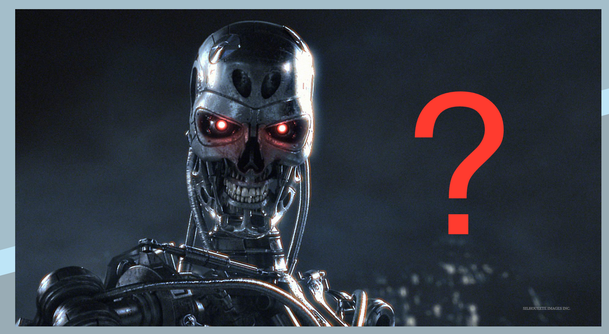








Conclusions



- Ultimately, we managed to build an object detection engine that extracts two features/transforms: blob_dog and hough_cirrcle.
- This feature extraction was the first step in the pipeline for classifying vesicles, and because we were not able to perform supervised learning on our vesicle detector, our precision/recall analysis would be incredibly poor and seemingly unnecessary to calculate because of the partial completion of our pipeline.
- Our future work will involve finishing building the remainder of the pipeline depicted in the middle figure by leveraging the sciki-learn library, and in particular, random forests in order to run supervised learning. This training would be done on a ground truth image of vesicles as noted by the black image directly to the left.

Acknowledgements

We'd like to thank all the people who gave there time to Introduction to Connectomics this intersession including all of our guest speakers but especially William Gray Roncal and Greg Kiar.

References and Further Information

William Gray Roncal, Michael Pekala, Verena Kaynig-Fittkau, Dean M Kleissas, Joshua T Vogelstein, Hanspeter Pfister, Randal Burns, R Jacob Vogelstein, Mark A Chevillet and Gregory D Hager. VESICLE: Volumetric Evaluation of Synaptic Interfaces using Computer Vision at Large Scale. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 81.1-81.13. BMVA Press, September 2015.