4th Project Final Report

Neuron Finder

Dharamendra Kumar
Pranjay Patil
Shubhi Jain
Yash Shrivastava

Masters in Computer Science

Instructor: Dr. Shannon Quinn

Department of Computer Science
University of Georgia, Athens

Project Specifications:

Language

- Python 2.7
- Apache Spark 2.0
- Thunder 1.4

Techniques

- Motion Correction
- Negative Matrix Factorization
- Source Extraction

Introduction

Calcium imaging is an extremely useful technique for investigating the variety of roles that calcium ions have in functioning neurons. Calcium ions generate a multitude of intracellular signals that control key functions, such as the neurotransmitter release from synaptic vesicles.

Problem Statement

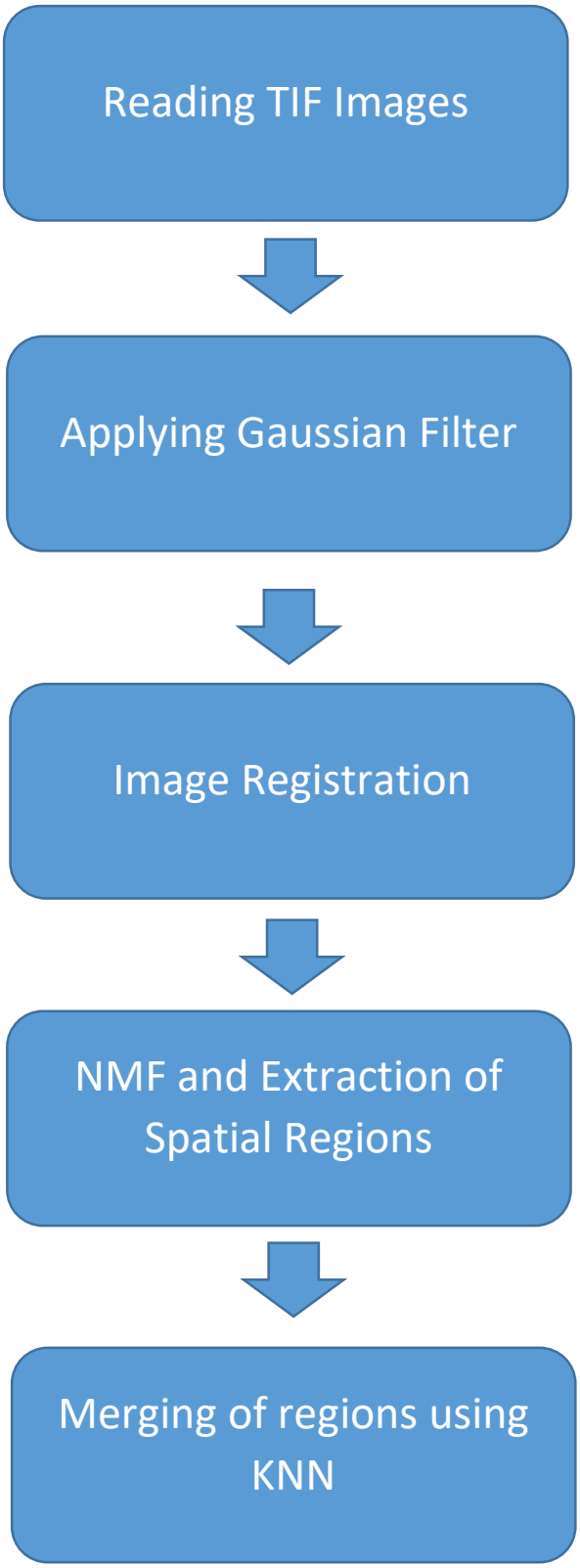
The problem is identifying neurons in a time-series image dataset. This can best be described as object-finding or image segmentation. In these, the goal is to design a model whose outputs are the coordinates to regions of interest within an image.

Approach

When it comes to image segmentation or object recognition, especially in the field of Calcium imaging, it is advisable to use Negative Matrix Factorization. There are many other ways to come up with a solution, i.e. convoluted 2D networks; however, for this project we chose to work with only one approach, using NMF.

Implementation

We referred to the GitHub repo (https://github.com/agiovann/Constrained_NMF) to get an idea of where to start. We used Apache Spark and Thunder for the implantation. We also contacted the developers of the Thunder package to discuss the algorithms that are implemented, and how to tweak the hyper-parameters for better accuracy. The following is a High Level block Diagram:



Reading TIF Images

The Neuro Finder website itself provides scripts that extract data from the website and load it to the local system. However, we downloaded all the datasets on a local machine manually and loaded the images into the program using Thunder API in a distributed fashion.

Gaussian Filter

After loading the images, we applied the Gaussian filter on images available in Python Scipy API for Noise reduction. Gaussian filter is 2-D convolution operator that is used to blur images, remove detail, and noise.

Image Registration

Image registration is the process of aligning two or more images of the same scene. This is also known as Motion Correction. Image registration is an important pre-processing technique for images, especially if the dataset consist of images that are taken at different times. We used the thunder-registration package that implements a set of registration algorithms. This package includes a collection of algorithms for image registration. It is well-suited to registering movies obtained in the medical or neuroscience imaging domains, but can be applied to any image sequences requiring alignment.

NMF and Extraction of Spatial Region

NMF, as stated before, was used to extract the spatial component of the images, which was further used to find the regions of interest. We used the NMF algorithm implementation available in thunder-extraction package.

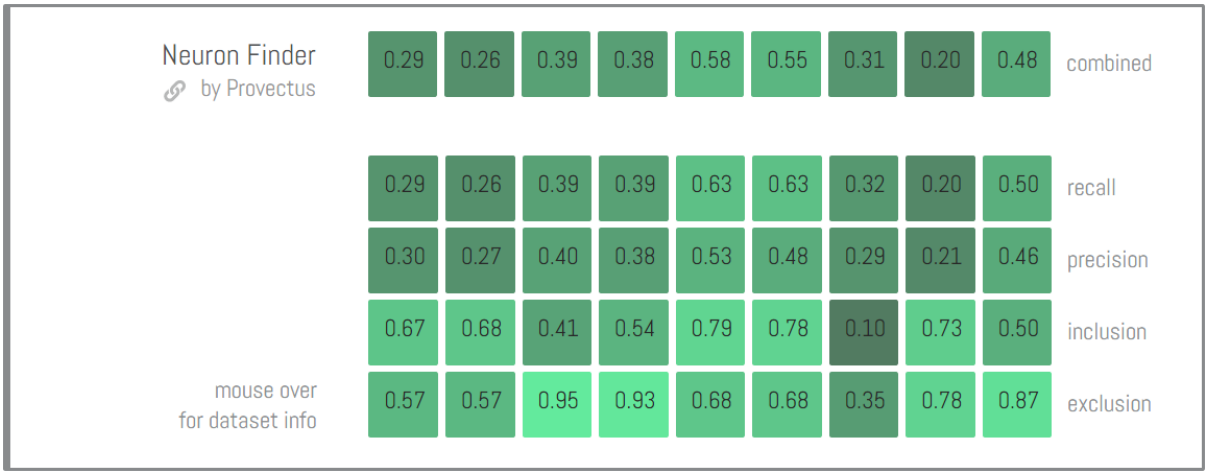
Merging of regions using KNN

The last step was to merge all the overlapping regions. This was achieved using the algorithm KNN. This is a greedy algorithm in which each region is merged with regions that have a high overlap. Only the k nearest neighbours are considered at each step, which dramatically speeds up the algorithm. The procedure concludes after making one or more iterations over all remaining regions.

Results

The output file is a JSON file that includes regions and its corresponding coordinates (pixels).

Following is the screen-shot of the results that we achieved after submitting the output file on the Neuro Finder Website (http://neurofinder.codeneuro.org).



You can also refer to NeuronFinder_Hyperparametrs.xlsx file on GitHub to read about various Hyper-parameters that we chose for different datasets.

References

1. Referred GitHub repo - https://github.com/agiovann/Constrained_NMF
2. NMF - https://arxiv.org/pdf/1409.2903v1.pdf
3. KNN - https://github.com/thunderproject/thunderextraction/blob/master/extraction/model.py
4. Thunder Package - http://thunder-project.org