Yale University
EliScholar – A Digital Platform for Scholarly Publishing at Yale

Yale Day of Data | Day of Data 2018

# Analyzing Neuronal Dendritic Trees with Convolutional Neural Networks

Olivier Trottier
*Yale University*, olivier.trottier@yale.edu

Jonathon Howard
*Yale University*, jonathon.howard@yale.edu

Follow this and additional works at: https://elischolar.library.yale.edu/dayofdata

Part of the Artificial Intelligence and Robotics Commons, Biophysics Commons, and the Developmental Biology Commons

Trottier, Olivier and Howard, Jonathon, "Analyzing Neuronal Dendritic Trees with Convolutional Neural Networks" (2019). *Yale Day of Data*. 5.
https://elischolar.library.yale.edu/dayofdata/2018/posters/5

This Event is brought to you for free and open access by EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in Yale Day of Data by an authorized administrator of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.

# Analyzing Neuronal Dendritic Trees with Convolutional Neural Networks

Olivier Trottier[1], Jonathon Howard[2]
[1]Department of Physics, [2]Department of Molecular Biophysics and Biochemistry, Yale University

## Question

How can we use neural networks to analyze neuronal dendritic trees?

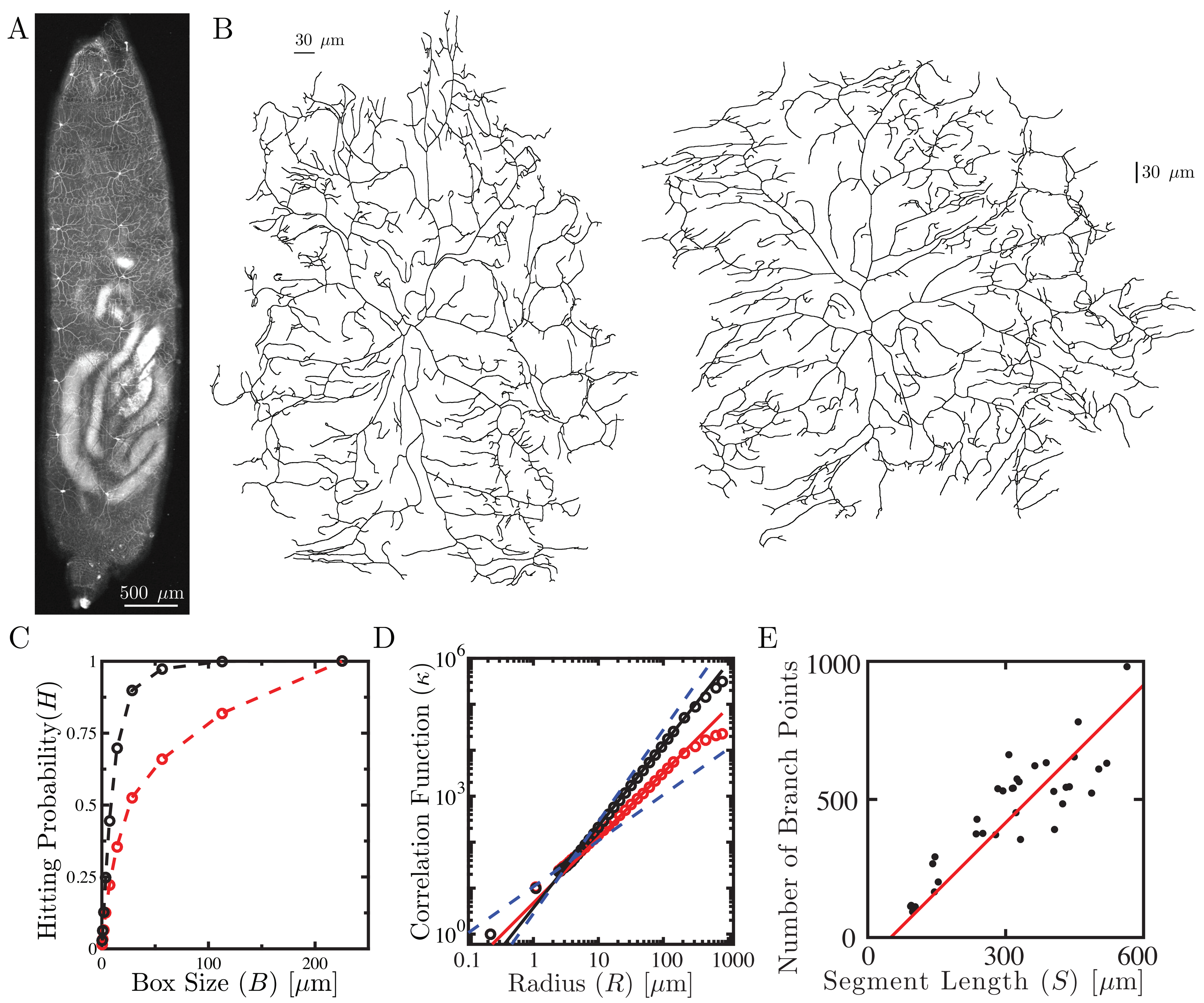
## Motivation

In the biological sciences, image analysis software are used to detect, segment or classify a variety of features encountered in living matter. However, the algorithms that accomplish these tasks are often designed for a specific dataset, making them hardly portable to accomplish the same tasks on images of different biological structures. Recently, convolutional neural networks have been used to perform complex image analysis on a multitude of datasets. While applications of these networks abound in the technology industry and computer science, use cases are not as common in the academic sciences. Motivated by the generalizability of neural networks, we aim to develop an algorithm based on neural networks to automatize image analysis of neuronal dendritic trees.

## Specific Goal

Detect branch points and branch tips in class IV dendritic trees of *Drosophila Melanogaster*.

## Class IV Dendritic Trees

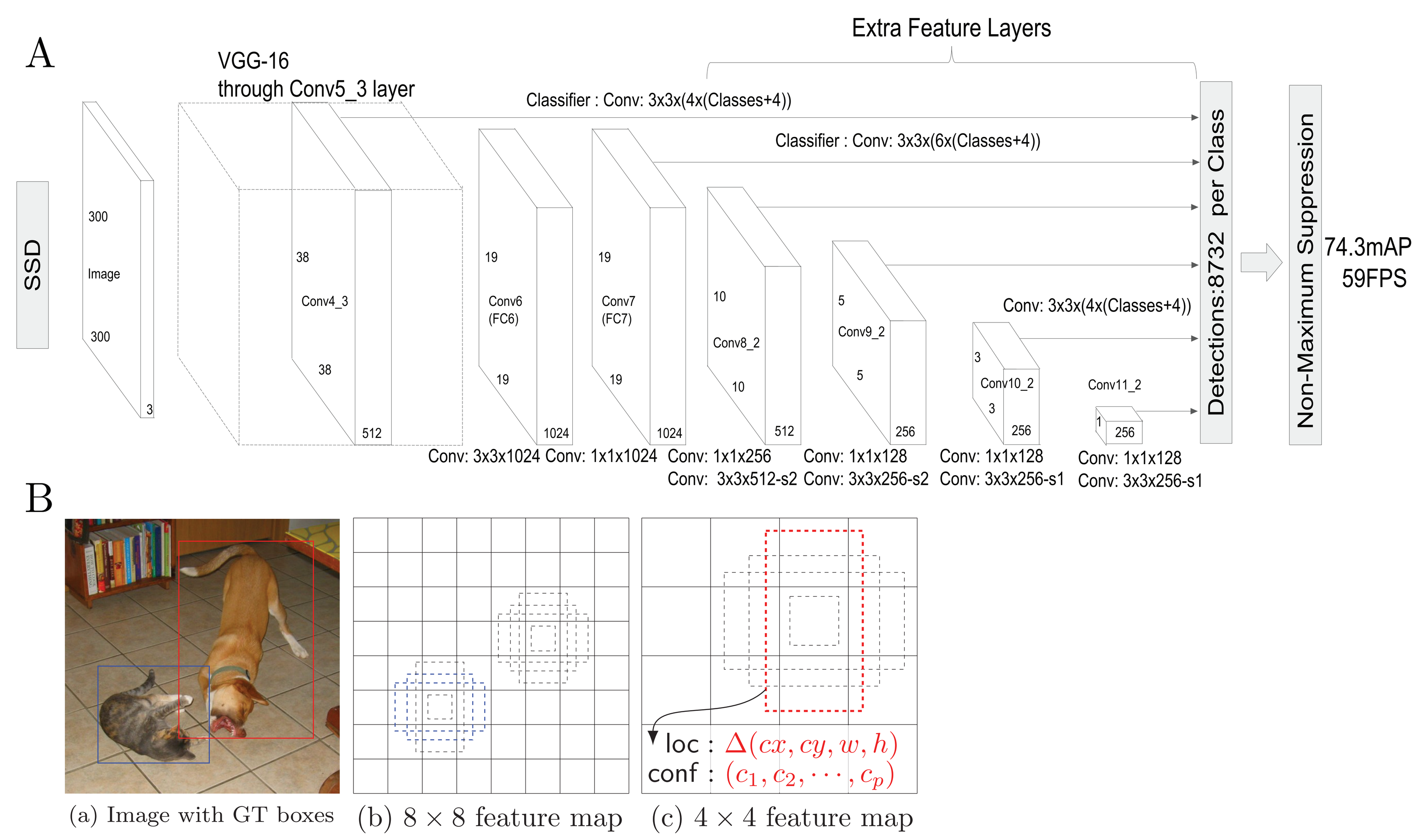



A) 3rd instar larval with the membrane of the class IV neurons labeled with GFP (ppk-cd4-tdGFP). The dendrites of the class IV neuron are nearly planar (500 × 500 × 10 μm).
B) Examples of class IV neuron skeletons recovered from software like Neuron Studio and ImageJ.
C)-E) Various metrics are calculated from the neuron skeletons to quantify the morphology of the dendritic trees. The hitting probability (C) assesses the ability of the shape to detect hits of a given size. The mass correlation function (D) is used to measure the fractal dimension. The number of branch points (E) quantify the hierarchy of the tree.

## Neural Network Model

To detect branch points and branch tips, we use the Single Shot Multibox Detector (SSD) proposed by *Liu et. al.*

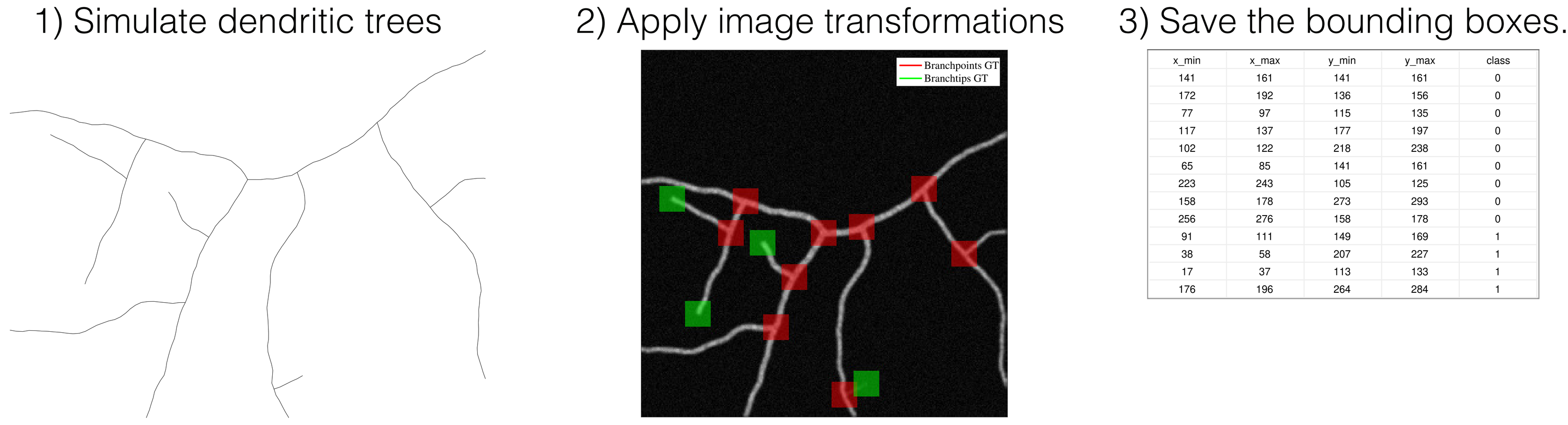
**Single Shot Multibox Detector**




A) Overview of the neural network architecture. The Single Shot Multibox Detector (SSD) is based on Visual Geometry Group (VGG) network, which consists of a sequence of 3x3 convolutional and max pooling layers. On the VOC2007 dataset, the network achieves an average precision of ~75%.
B) a) Examples of the grounding bounding boxes for the dog and cat classes. b) Example of a feature map that tiles the image with a set of default boxes, which serve as reference points for predicting the positions of sizes of the true bounding boxes. c) The final loss function combines the location loss and a class confidence loss of 8732 default boxes per class.
Figures A) & B were imported from *Liu et. al.*

## Training Dataset

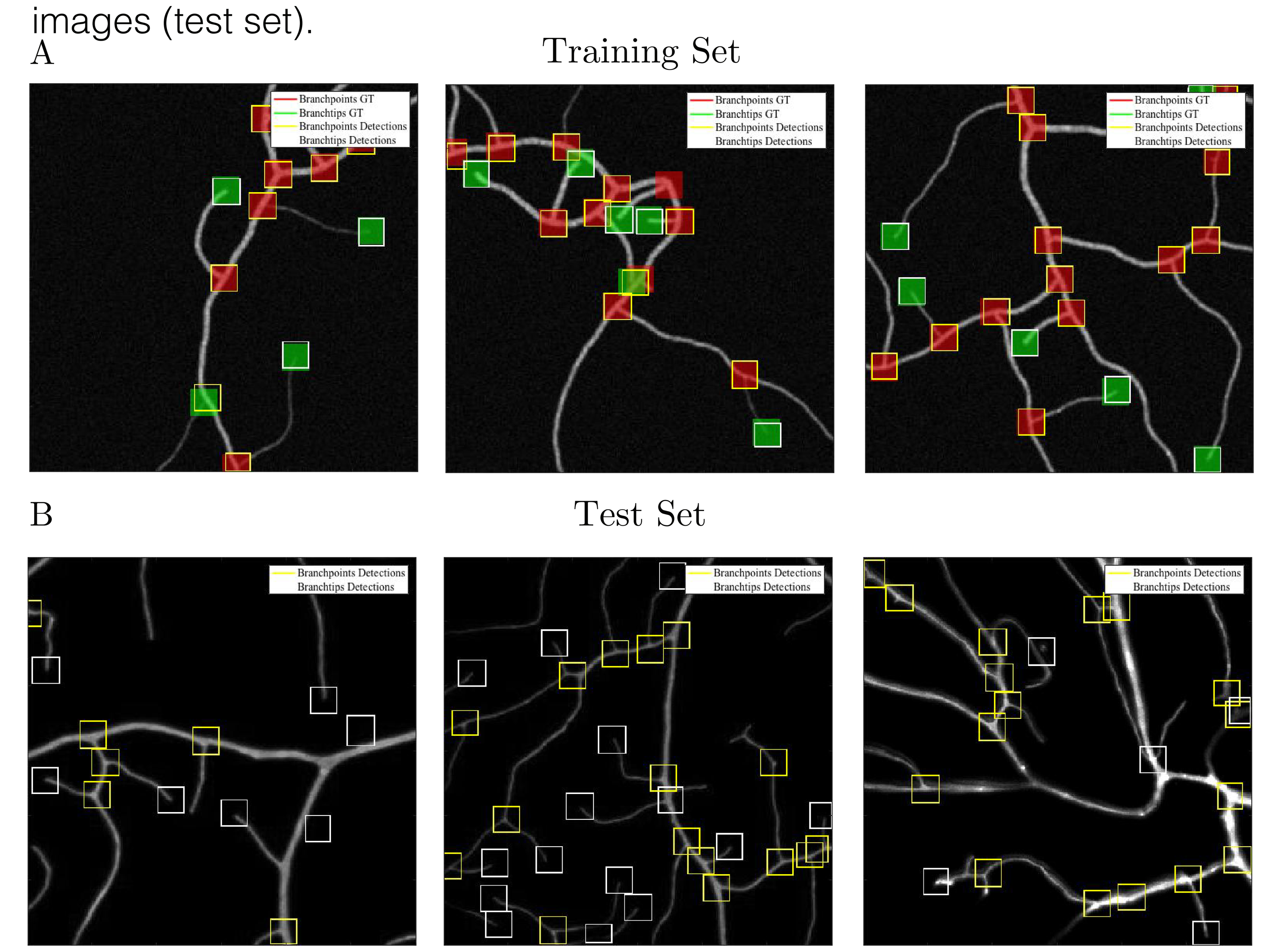
To train the neural network, we synthesize a set of dendritic trees using an algorithm that we previously developped.




A) Workflow for synthesizing the training set. First, we simulate dendritic trees using an algorithm that was previously implemented. Second, we perform various transformations such as rasterization, filtering and noise addition to convert the simulated skeletons into real-looking images. Finally, we save the bounding boxes position for branch points (0) and branch tips (1) as given by the simulations.

## Results

Our preliminary results indicate that the neural network learns to detect objects accurately on the training set, but fails to detect all objects on real images (test set).




A) Detections on the training set. Shaded squares represent ground truth bounding boxes while hollow rectangles represent detected bounded boxes.
B) Detections on the test set. In some cases, the neural network performs very well on detecting objects, but fails in cases where the image's pixel intensity has high variability.

## Future Directions

- Improve the training set synthesizer by implementing higher pixel intensity variations.
- Simplify the SSD architecture based on the dataset (reduce number of default boxes).
- Train with a different neural network architecture (RetinaNet).
- Perform live tracking on developmental movies.

## Acknowledgments

We thank Maijia Liao and Sonal Shree for providing us with confocal microscopy images of class IV dendritic trees. We also wish to thank the National Institute of Health (NIH) and the Integrated Program in Physical, Engineering and Biology (PEB) for financial support.

NATIONAL INSTITUTES OF HEALTH

PEB

## References

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. ECCV 2016, pages 21–37, Cham, 2016. Springer International Publishing.