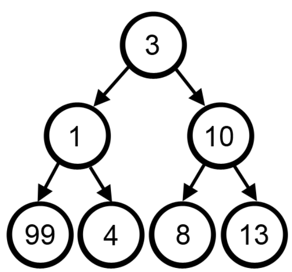
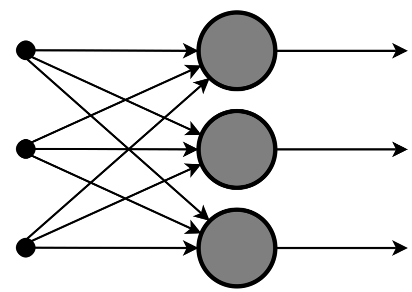
CSE 40171: Artificial Intelligence





Connectomics: Flood Filling Networks for the Segmentation of Neural Volumes

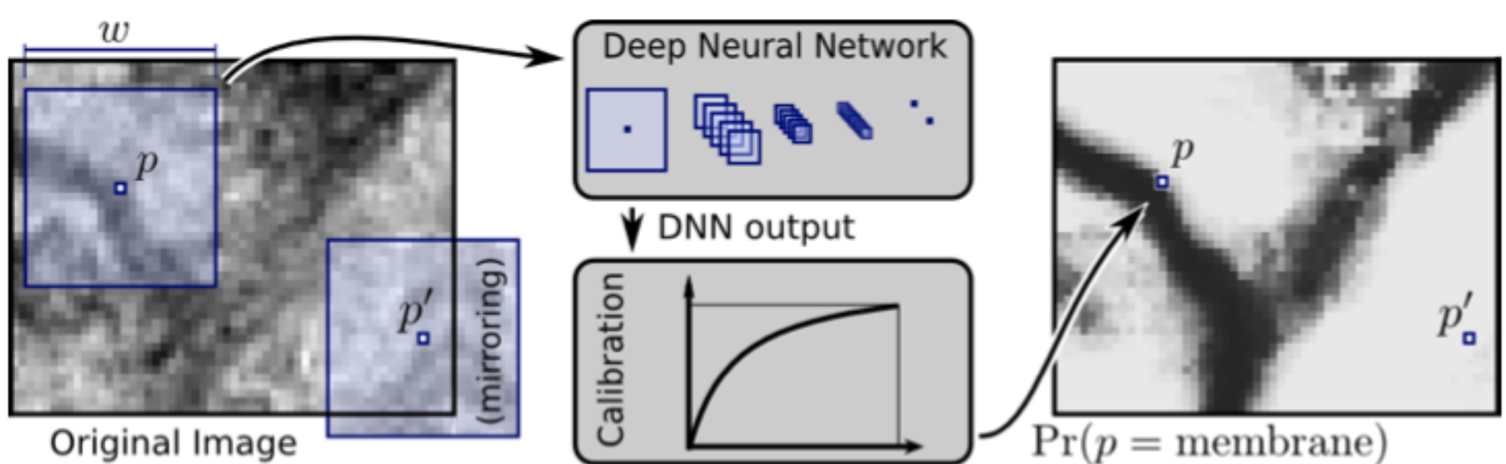
Homework #5 has been released
It is due at 11:59PM on 11/13

Project Updates are Due on 11/25 at 11:59PM

(See Course Website for Instructions)

What advantages do we get applying deep learning for segmentation?

Early Attempt at Neural Network Pixel Classification: D. Cirşan et al. NeurIPS 2012



Handle edge effects

DNN Architecture

- Each **Convolutional Layer** performs a 2D convolution of its input maps with a square filter
    - Sum the convolutional responses, which are passed through a non-linear activation function
- **Max-Pooling** selects the most promising features out of the non-overlapping square regions
- **Fully Connected Layers** combine the outputs into a 1D feature vector
    - Using a softmax activation function for the last layer guarantees an interpretation of a probability of a particular input image belonging to a class

Training

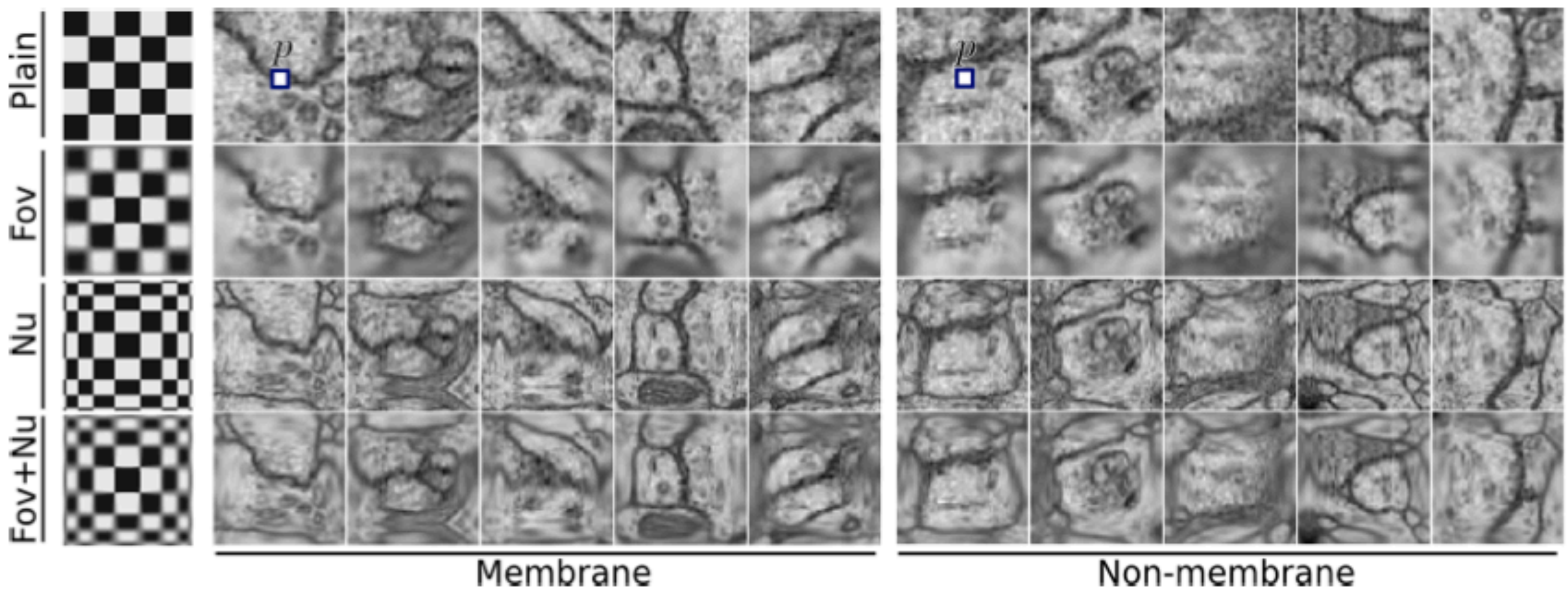
- All available slices from the ISBI training stack
  - 30 images, 512x512 resolution
- For each slice:
  - all membrane pixels are treated as positive examples (~50,000 per slice)
  - same number of pixels is randomly sampled from non-membrane pixels for negative examples
- Three million total training examples (balanced +/-)

Post-processing

- Classes are represented equally in training set
  - Severe overestimation of membrane probability
  - Fix: polynomial function post-processor is applied to network outputs
- Calibration function is defined by training a network $\mathbf{N}$ on 20 slices from training volume $\mathbf{T}_{\text{train}}$ and testing on the remaining 10 slices $\mathbf{T}_{\text{test}}$
  - Compare all output from $\mathbf{T}_{\text{test}}$ to ground truth
  - Compute transformation relating output value from $\mathbf{N}$ and the actual probability of being membrane
    - well approximated by monotone cubic polynomial

Foveation and Nonuniform Sampling

Additional Pre-processing



Averaging Outputs of Multiple Networks

Large networks with different architectures exhibit significant output differences

Reduce variance by averaging calibrated outputs of different networks

Fusion is better than the performance of any single network

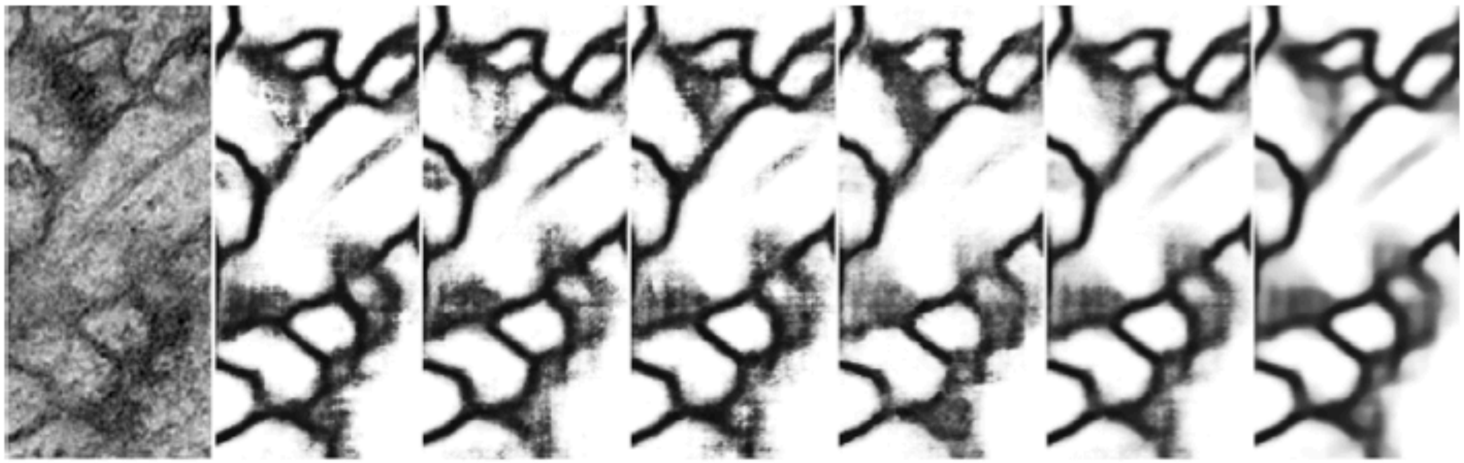
What does a network look like?

N4, $w = 95$

Layer	Type	Maps and neurons	Kernel size
0	input	1 map of 95x95 neurons	
1	convolutional	48 maps of 92x92 neurons	4x4
2	max pooling	48 maps of 46x46 neurons	2x2
3	convolutional	48 maps of 42x42 neurons	5x5
4	max pooling	48 maps of 21x21 neurons	2x2
5	convolutional	48 maps of 18x18 neurons	4x4
6	max pooling	48 maps of 9x9 neurons	2x2
7	convolutional	48 maps of 6x6 neurons	4x4
8	max pooling	48 maps of 3x3 neurons	2x2
9	fully connected	200 neurons	1x1
10	fully connected	2 neurons	1x1

Experimental Results

Calculated error for four networks



Source		**N1** *w=65* Fov+Nu	**N2** *w=65* Fov+Nu	**N3** *w=65* Plain	**N4** *w=95* Fov+Nu	Averaged	Averaged +Filtered
Rand err.	$[\cdot 10^{-3}]$	64	68	57	61	**48**	
Warping	$[\cdot 10^{-6}]$	457	485	618	524	**434**	after filtering
Pixel err.	$[\cdot 10^{-3}]$	65	66	66	68	**60**	

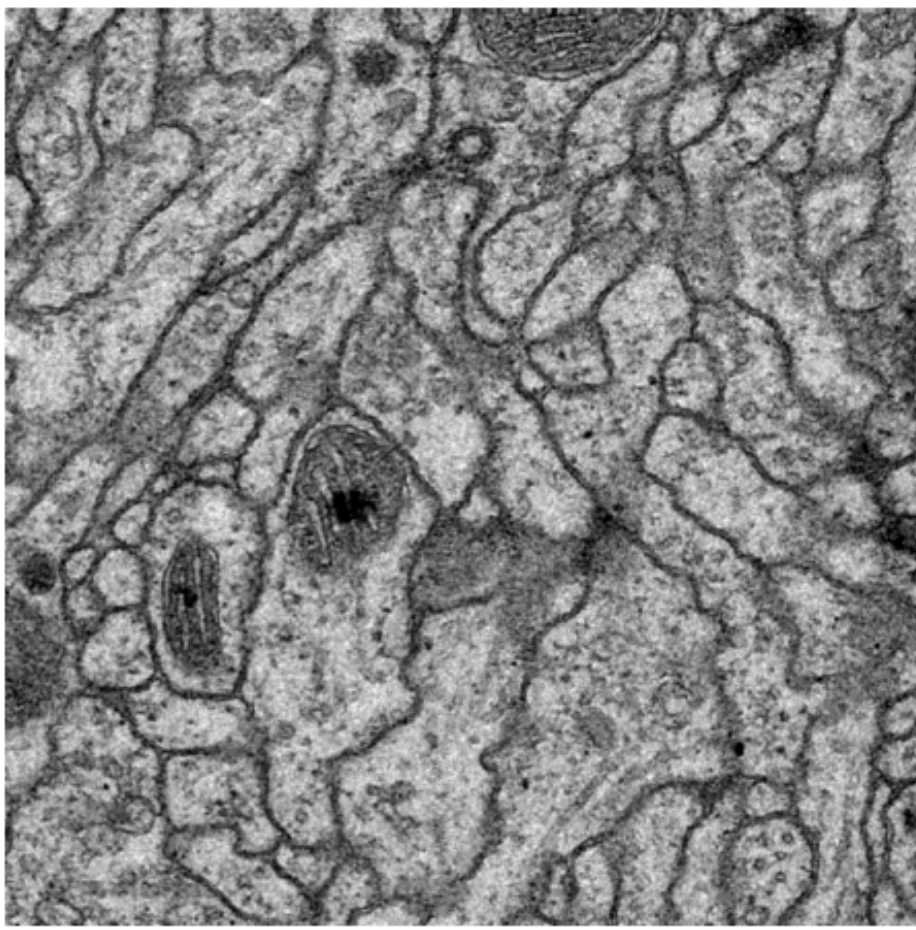
Experimental Results

ISBI 2012 Scores:

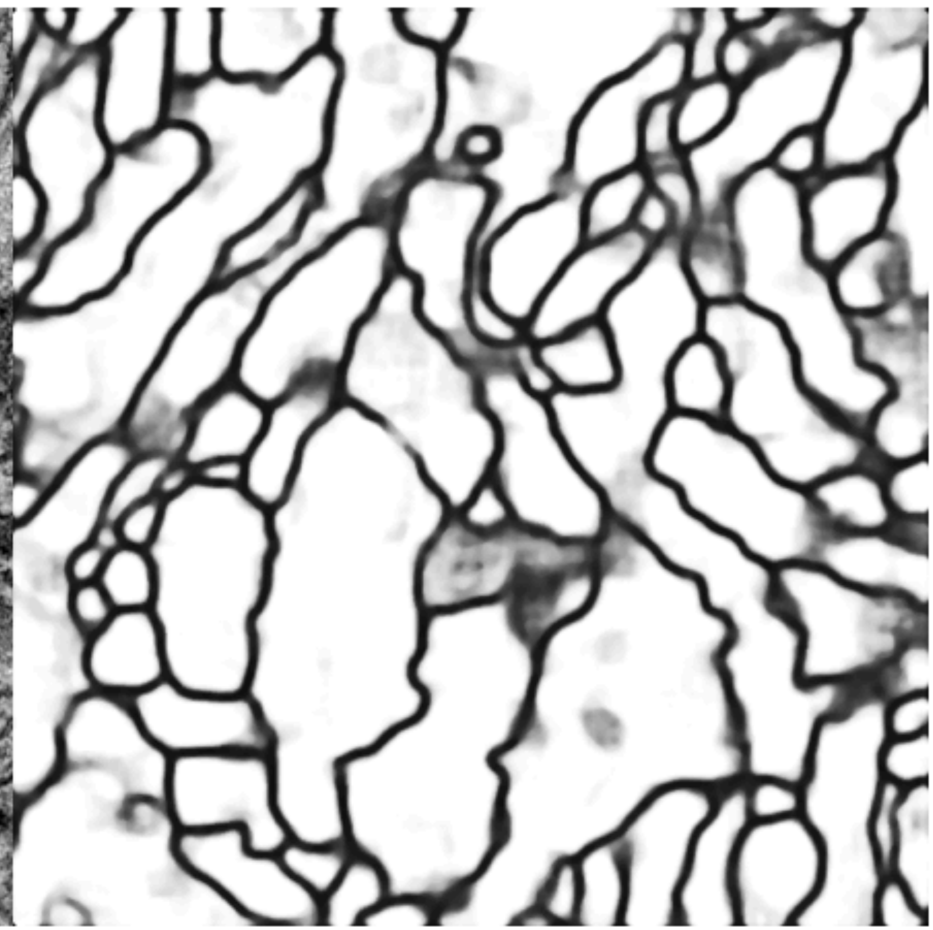
0.973015769 (Rand Score)

0.987441379 (Information Score)

Slice 16 of stack



Segmentation



What’s missing from this approach?

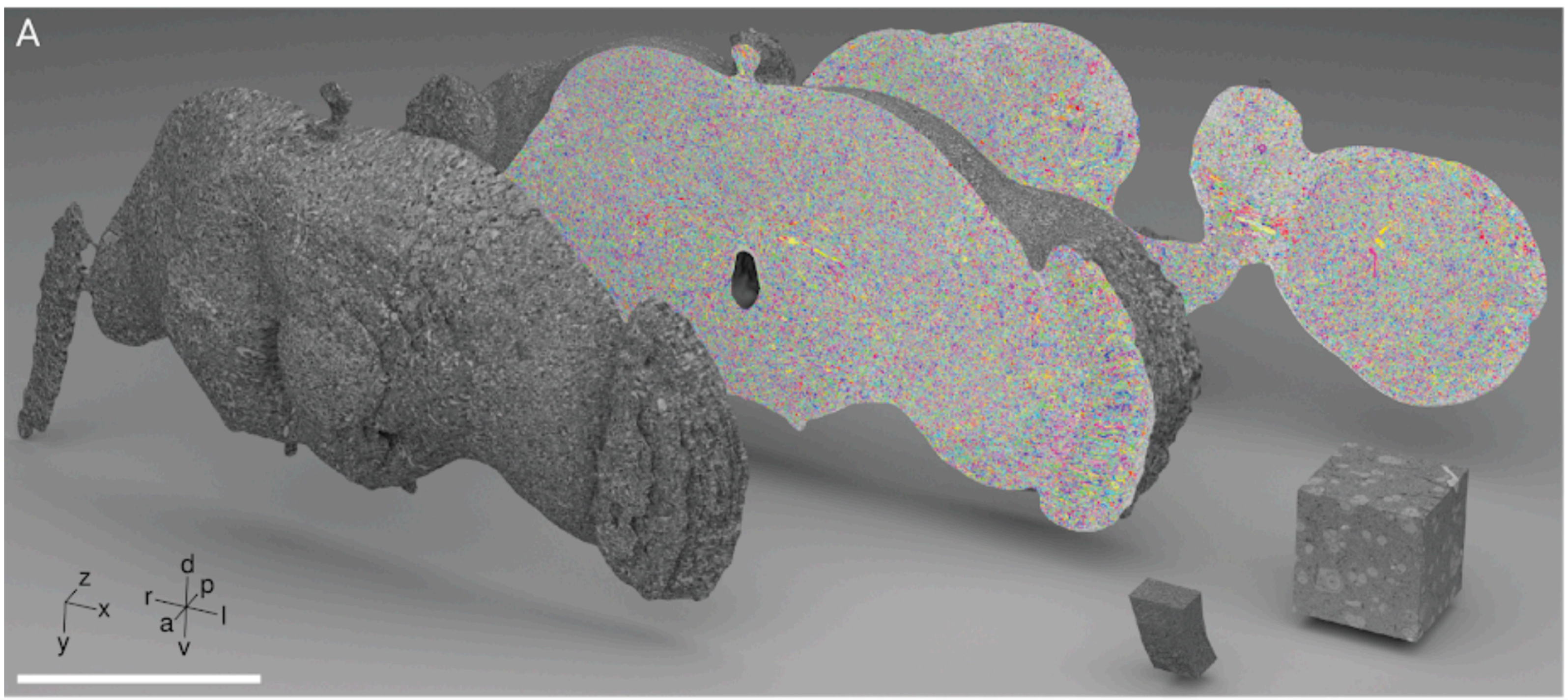
No pixel clustering to separate the neurites

Current Best Approach: Flood-Filling Networks (FFN)

Google's Contribution: Deep Learning + Hardware at a Massive Scale

V. Jain et al.

Strategy: create a unified neural network architecture that can scale to entire brains



P. Li et al. bioRrxiv 2019

Flood-Filling Networks (FFN):

- Januszewski et al. arXiv 2016
- Two successive and distinct computations in one NN architecture
  - pixel classification (membrane / non-membrane)
  - cluster pixels into segments that represent neurites ("flood filling")
- Recurrent 3D convolutional network
- https://github.com/google/ffn

Architecture of an FFN

Process sub-volumes containing **raw pixels** into individual object masks:

Strategy 1. CNN using “object mask channel” to both specify the target object and provide an explicit memory of segmentation state across recurrent iterations

Strategy 2. Recurrent procedure for iterating the network inference dynamics over multiple overlapping fields of view in order to segment arbitrarily large objects

Architecture of an FFN

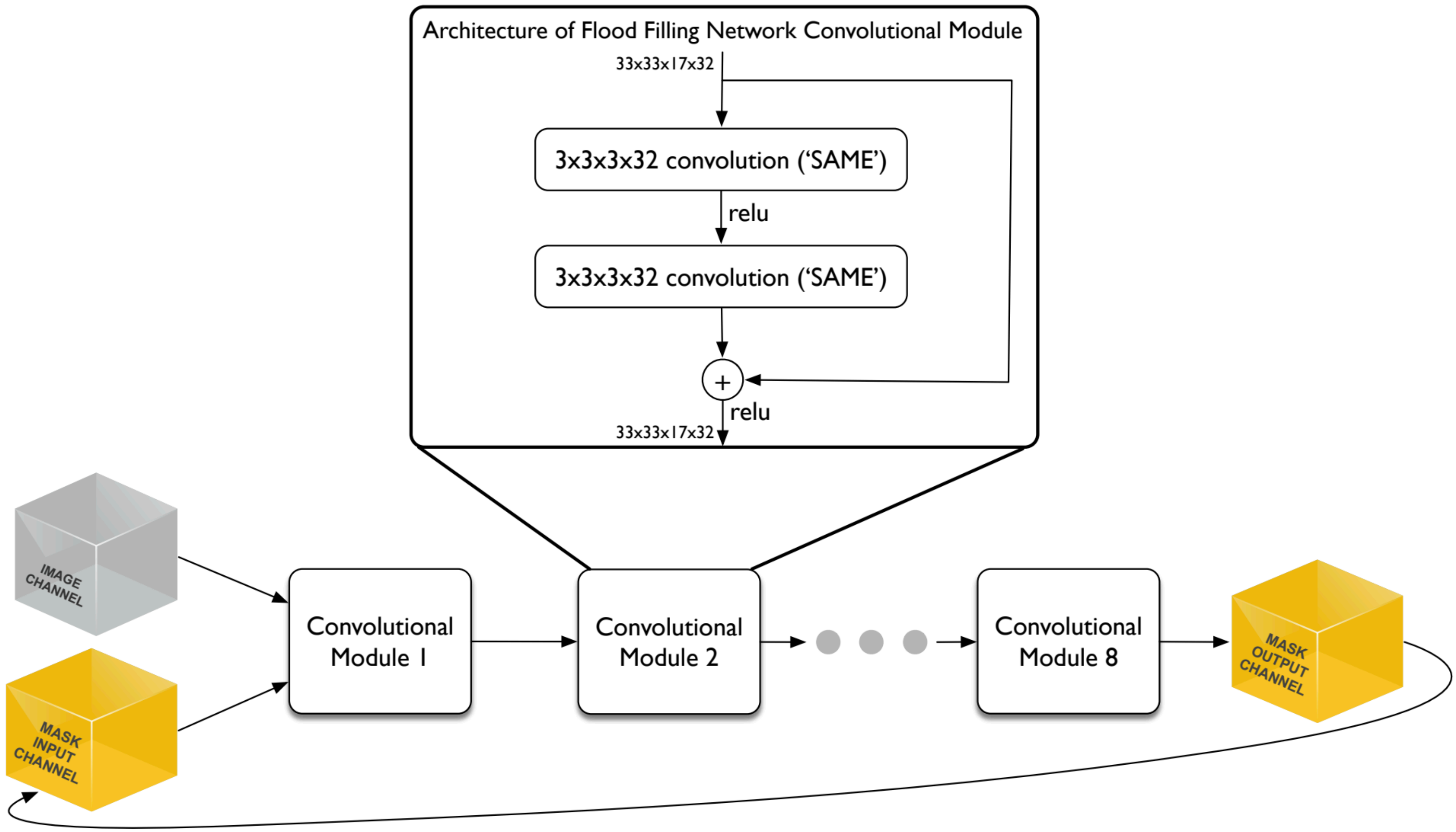
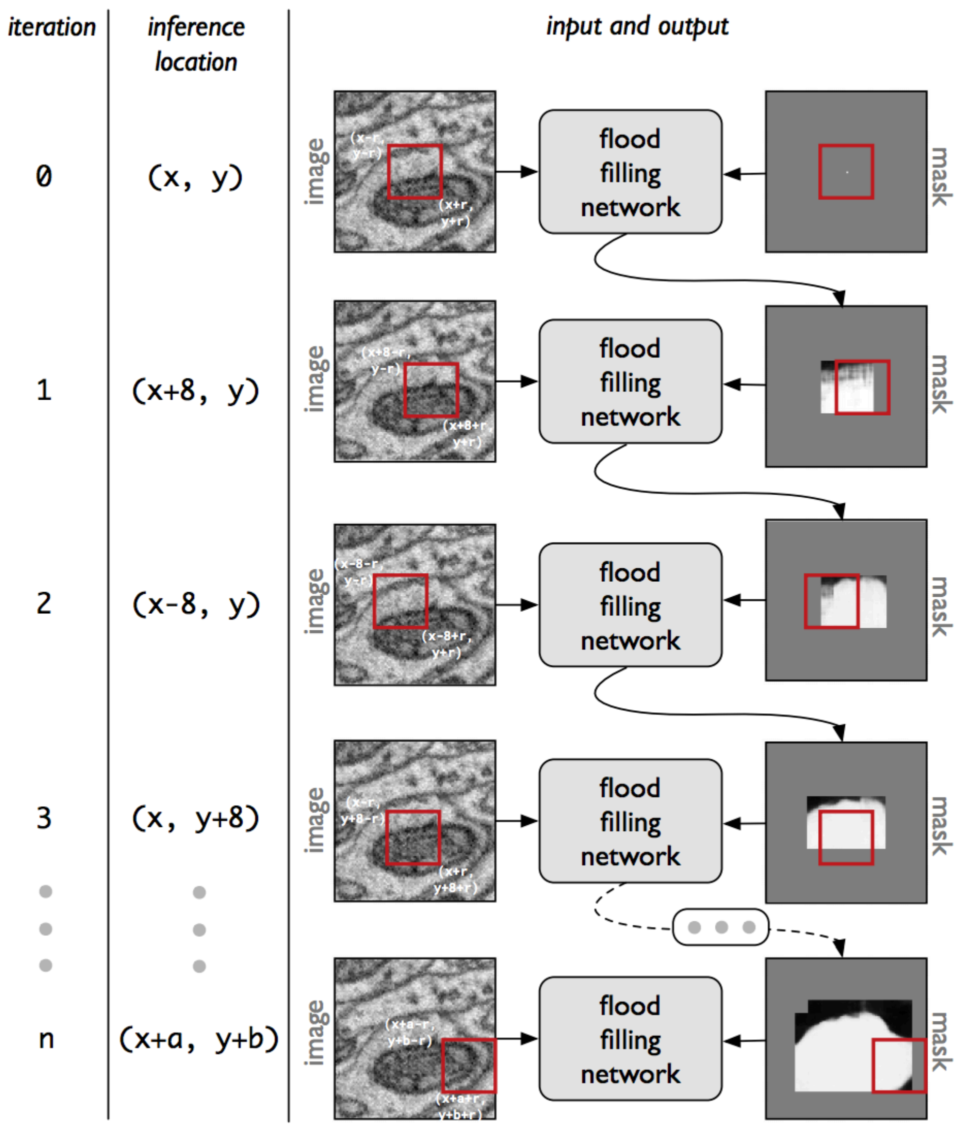


Image Credit: Januszewski et al. arXiv 2016

Multiple field-of-view inference of an FFN



Implementation

- Implemented in TensorFlow (https://www.tensorflow.org/)
- Trained with asynchronous gradient descent
- Distributed training using 32 NVIDIA K40 GPUs
- **33 densely annotated training volumes**

Dataset used to validate the approach

Neuropil of the zebra finch songbird (Taeniopygia guttata) Area X was imaged using Serial Blockface Scanning Electron Microscopy (SBEM) at $10 \times 10 \times 20$ nm

Out of the complete ~500-gigavoxel dataset, 29 150^3 voxel and four $256 \times 256 \times 128$ voxel spatially-separated volumes were manually annotated

A separate $520 \times 520 \times 256$ volume was densely skeletonized using Knossos (http://knossostool.org/) and used here as a testing set

- Within this volume, 221 disconnected fragments were skeletonized with a total of 5234 edges, corresponding to a path length of ~1 mm

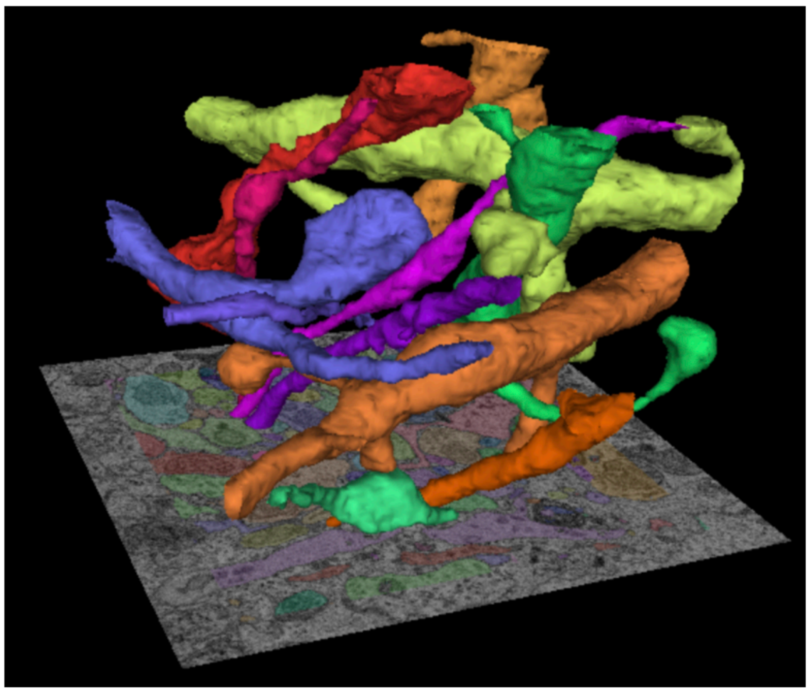
FFN Segmentation Performance

Edge accuracy on the 520×520×256 test volume

Segmentation	Edge accuracy [%]	Merged edges [%]	Split edges [%]	Omitted edges (adjusted) [%]	Omitted edges (raw) [%]
CNN + Watershed	87.7	1.0	10.6	0.7	1.1
CNN + Watershed + GALA	96.3	1.7	1.4	0.6	1.1
CNN + Watershed + CELIS	93.2	5.4	0.7	0.7	1.1
Flood-Filling Network	98.5	0.0	0.7	0.8	2.4

FFN Segmentation Performance

Mix of dendrites and axons



Fragments of three glial cells

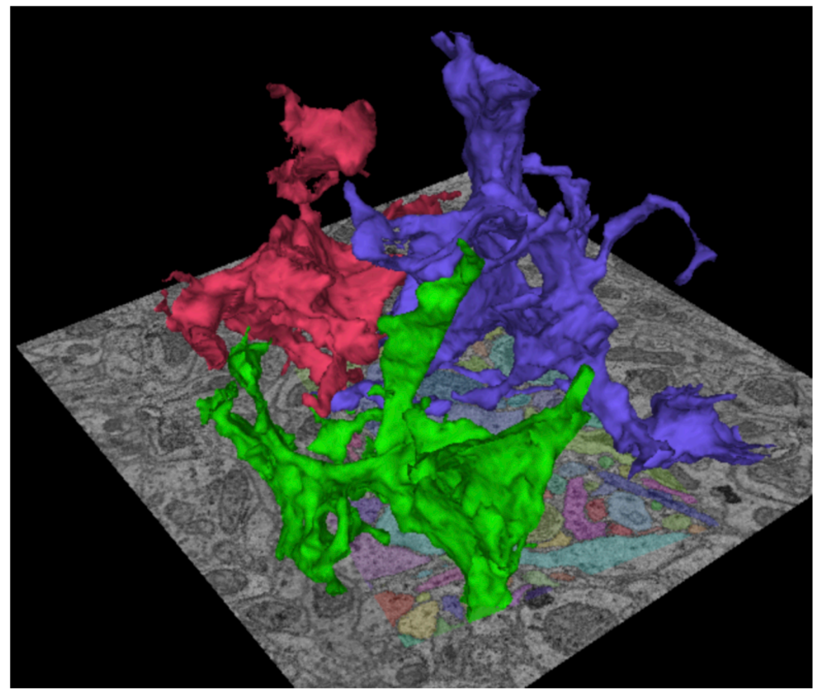


Image Credit: Januszewski et al. arXiv 2016

Computational Cost

Significantly increased computational cost (inference) caused by the depth of the network, and the single object segmentation scheme

For the densely skeletonized subvolume:

- the cost of the 3D affinity graph CNN inference is approximately 0.14 PFLOP
- total cost of the FFN segmentation is 4.6 PFLOP

Additional cost factor proportional to the number of distinct objects present within the FoV of the network

Application to fly brain

Segmentation of a forty-teravoxel whole-brain Drosophila ssTEM volume

Additional algorithmic component: re-alignment procedures

Largely merger-free segmentation of the entire ssTEM Drosophila brain

Drosophila melanogaster Proboscis

No longer a unified NN architecture

Now makes use of pre-processing strategies, like most other approaches:

Local Realignment corrects residual misalignment within each local subvolume block just prior to segmentation

Irregular Section Substitution addresses the problems of damaged, occluded, missing, and distorted areas by selectively replacing these areas with data from neighboring sections

Local Realignment

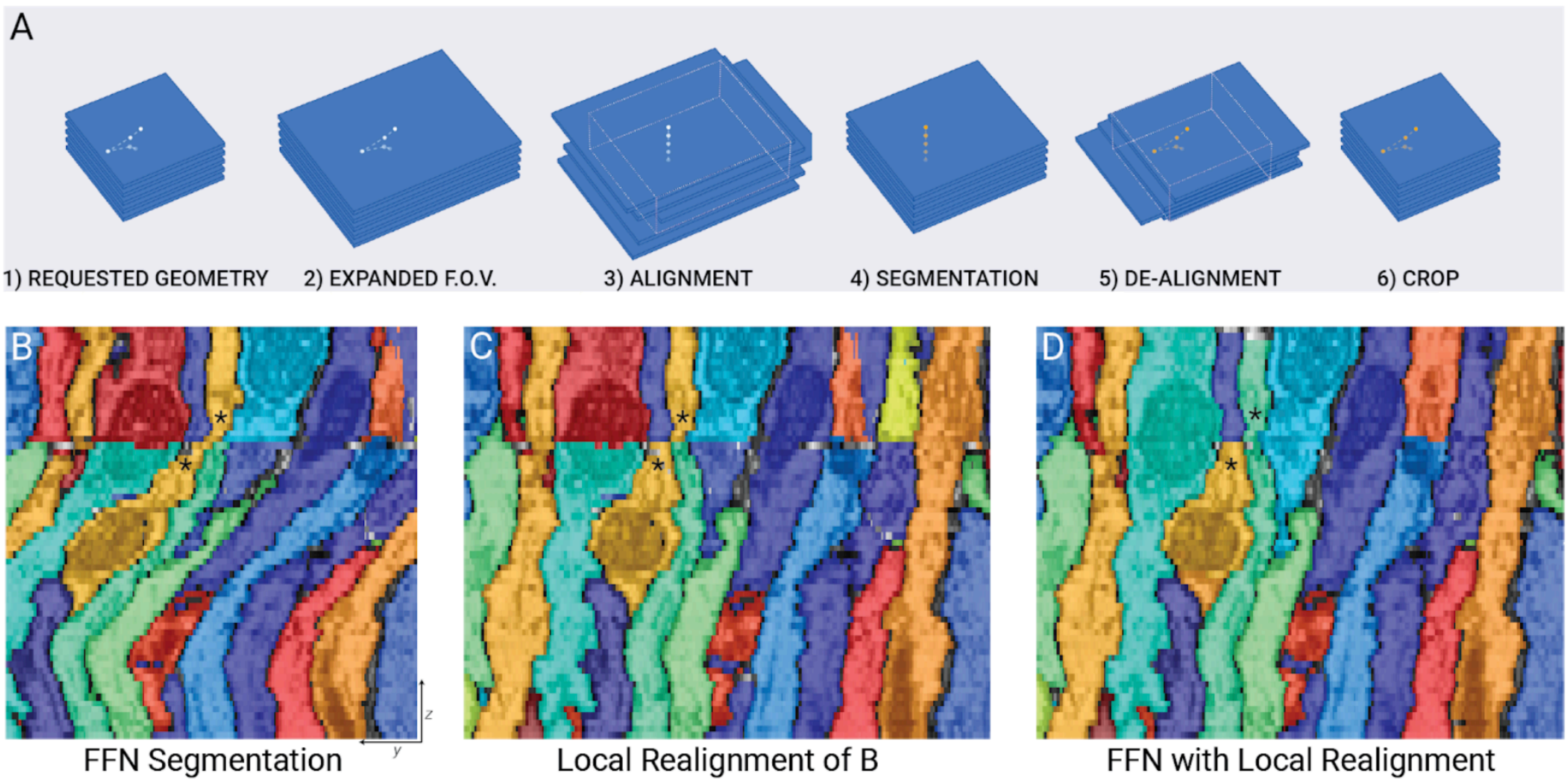
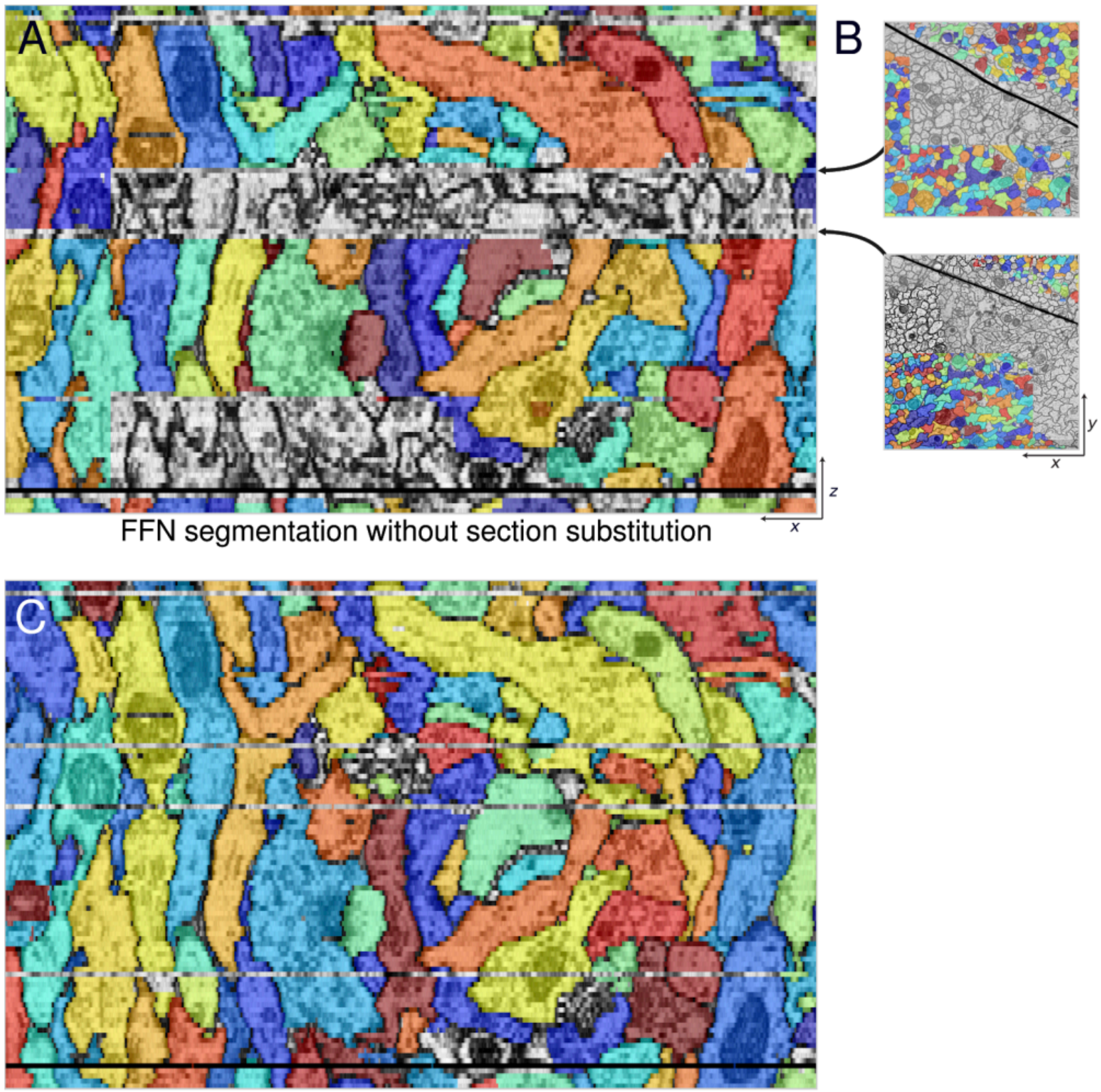


Image Credit: P. Li et al. bioRrxiv 2019

Irregular Section Substitution



FFN segmentation without section substitution

With irregular section substitution (ISS)

Image Credit: P. Li et al. bioRrxiv 2019

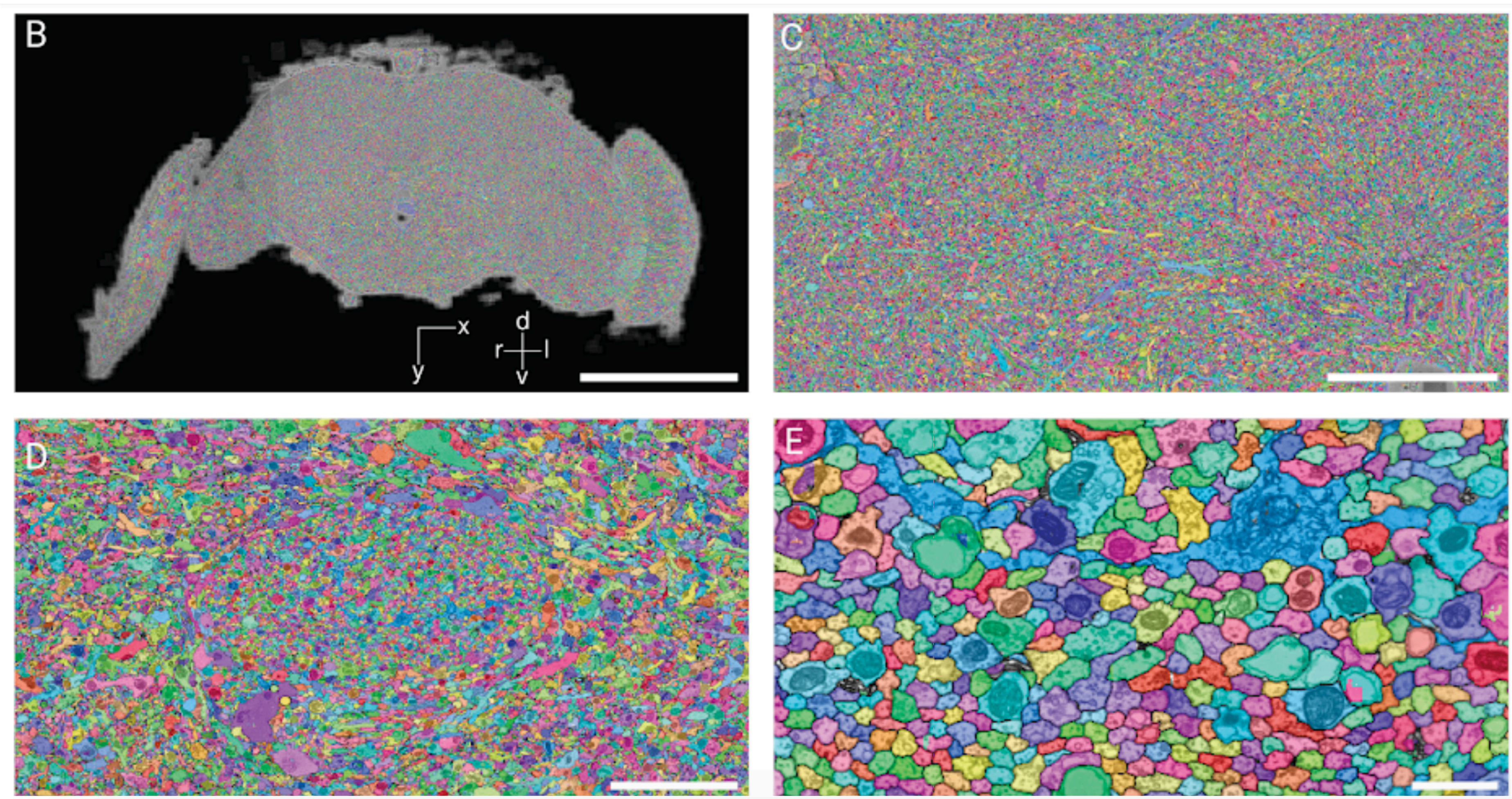
Input and Training Data

Raw image and training data were all derived from the Full Adult Fly Brain (FAFB) dataset described in Zheng et al. Cell 2018

Training data consisted of three densely labeled cutouts from the Mushroom Body region of an earlier FAFB global alignment

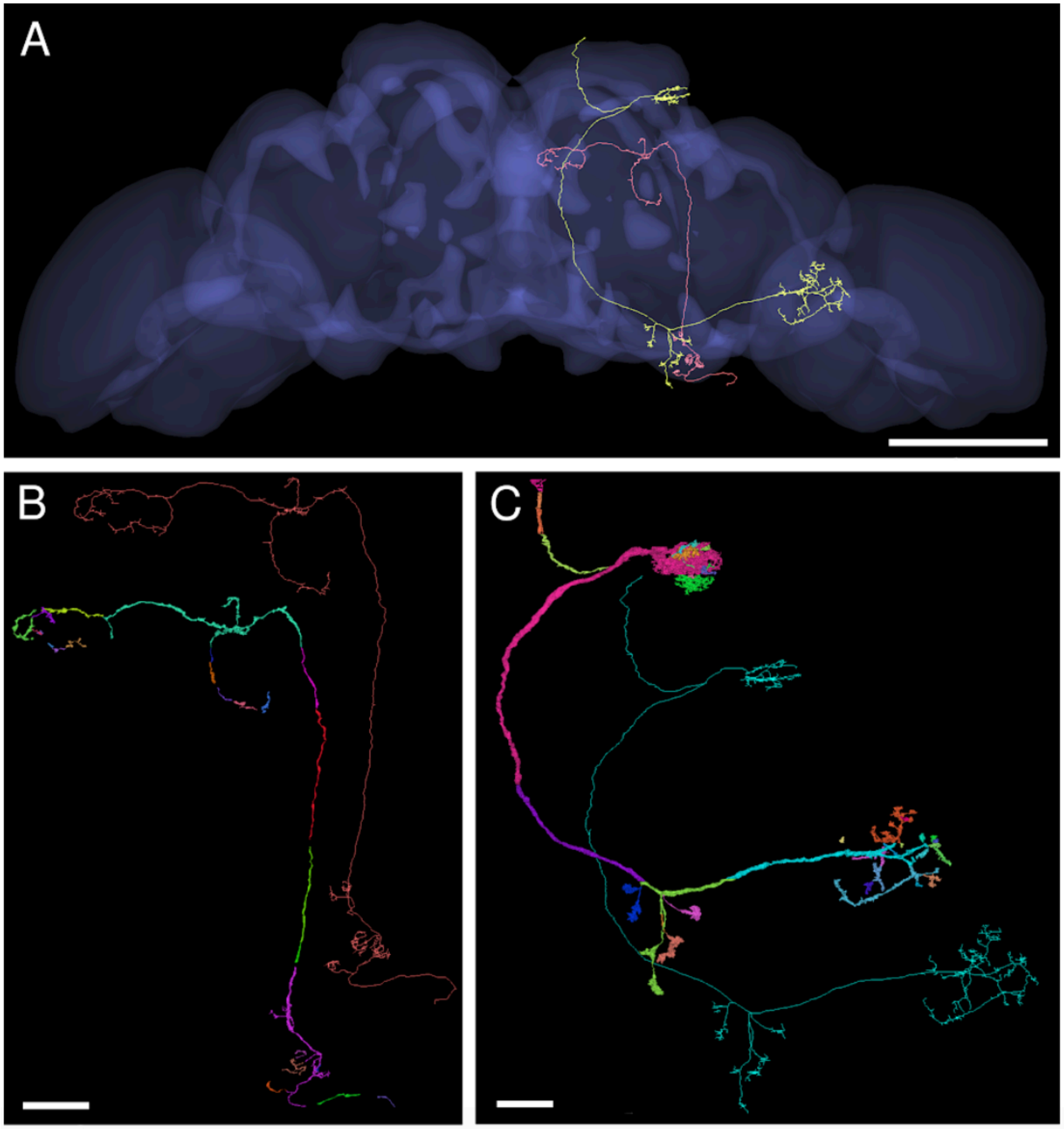
- MICCAI Challenge on Circuit Reconstruction from Electron Microscopy Images (CREMI)
- Each cutout totals 1250x1250x125 labeled voxels at 4x4x40 nm

Dense segmentation of entire fly brain



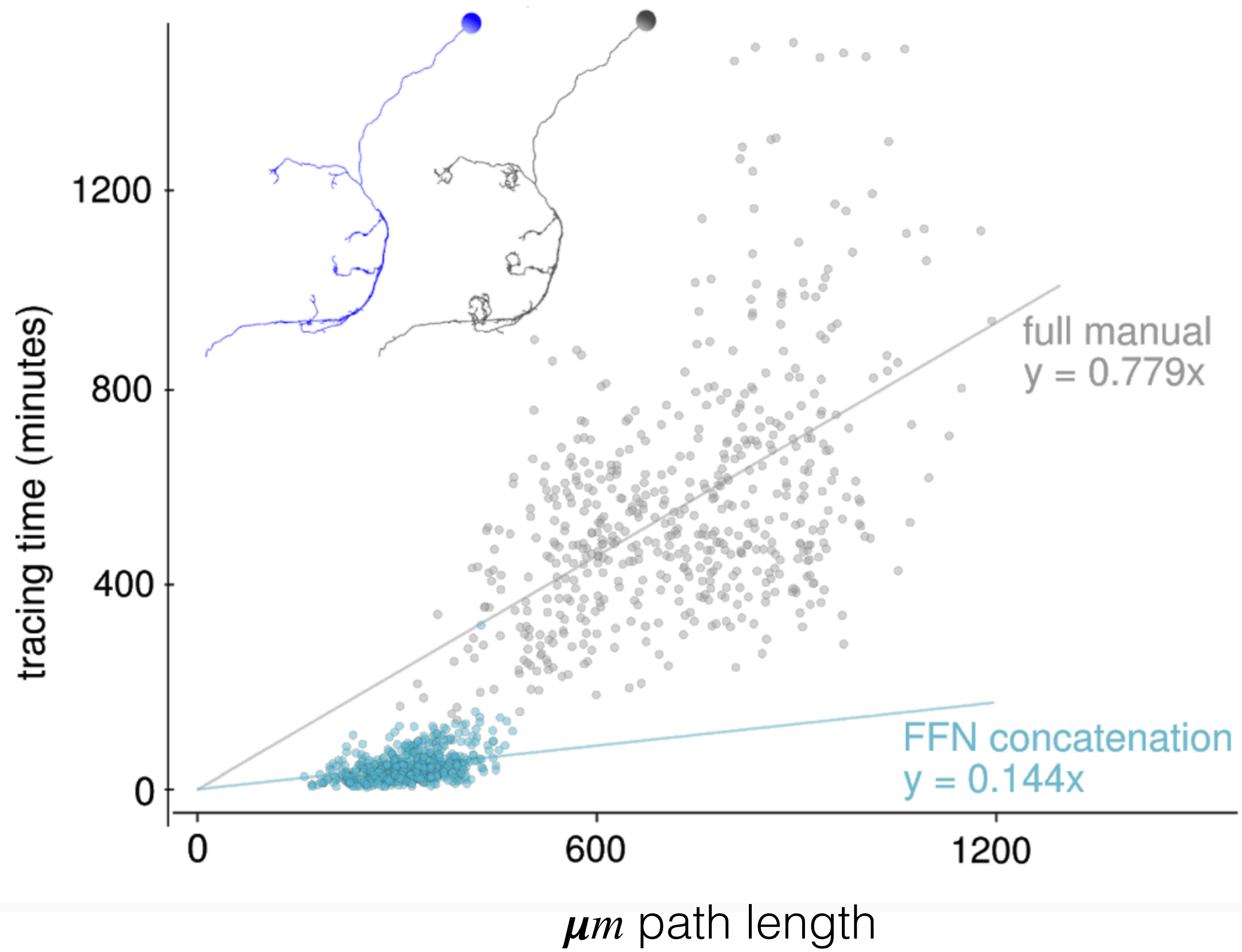
P. Li et al. bioRrxiv 2019

FFN vs. Ground Truth



P. Li et al. bioRrxiv 2019

Tracing Speed



P. Li et al. bioRrxiv 2019

Remaining Challenges

Split errors on long processes (e.g., axons)

Poor generalization across datasets, microscopes, animals

Difficulty obtaining ground-truth annotations

Is the signal-to-noise ratio even favorable to ML?