S. Soltanian-Zadeh , K. Sahingur , S. Blau, Y. Gong , and S. Farsiu,"Fast and robust active neuron segmentation in two-photon calcium imaging using spatio-temporal deep-learning", (IN PRESS), Proceedings of the National Academy of Sciences (PNAS), 2019
https://doi.org/10.1073/pnas.1812995116

ABSTRACT

Calcium imaging records large-scale neuronal activity with cellular resolution *in vivo*. Automated, fast, and reliable active neuron segmentation is a critical step in the analysis workflow of utilizing neuronal signals in real-time behavioral studies for discovery of neuronal coding properties. Here, to exploit the full spatio-temporal information in two-photon calcium imaging movies, we propose a three-dimensional convolutional neural network to identify and segment active neurons. By utilizing a variety of two-photon microscopy datasets, we show that our method outperforms state-of-the-art techniques and is on a par with manual segmentation. Furthermore, we demonstrate that the network trained on data recorded at a specific cortical layer can be used to accurately segment active neurons from another layer with different neuron density. Finally, our work documents significant tabulation flaws in one of the most cited and active online scientific challenges in neuron segmentation. As our computationally fast method is an invaluable tool for a large spectrum of real-time optogenetic experiments, we have made our open-source software and carefully annotated dataset freely available online.

SIGNIFICANCE

Two-photon calcium imaging is a standard technique of neuroscience labs that records neural activity from individual neurons over large populations in awake-behaving animals. Automatic and accurate identification of behaviorally-relevant neurons from these recordings is a critical step towards complete mapping of brain activity. To this end, we present a fast deep-learning framework which significantly outperforms previous methods and is the first to be as accurate as human experts in segmenting active and overlapping neurons. Such neuron detection performance is crucial for precise quantification of population-level and single-cell level neural coding statistics, which will aid neuroscientists to temporally synchronize dynamic behavioral or neural stimulus to the subjects' neural activity, opening the door for unprecedented accelerated progress in neuroscientific experiments.

INTRODUCTION

Advances in two-photon microscopy and genetically encoded calcium indicators have enabled high-speed and large-scale *in vivo* recording of neuronal populations at 5-60 Hz video rate data (1-5). Fast, automatic processing of the resulting large imaging datasets is a critical yet challenging

step for discovery of neuronal coding properties in behavioral studies. Often the investigators are interested in identifying a subset of active neurons from the large imaged population, further complicating the neuronal segmentation task. The subset of modulating, and thus active, neurons in many behavioral experiments carry the meaningful information for understanding the brain's coding characteristics. Automatic identification of active neurons from the imaging movies in high speed enables scientists to directly provide dynamic complex behavioral or neural stimulus to the subjects in real-time.

Recent efforts from several groups have produced automatic methods to detect and quantify neuronal activity in calcium imaging data. These methods span from unsupervised classic machine learning techniques (6-16) to deep-learning based supervised algorithms (17, 18). Among the former class of neuron segmentation algorithms are the popular methods of principal component and independent component analysis (PCA/ICA) (11), constrained non-negative matrix factorization (CNMF) (13), extension of CNMF to one-photon microscopy (16), and the more recent and faster version of CNMF, called OnACID (7), which is based on online dictionary learning. Recently, Giovannucci et al. (19) have improved the scalability of CNMF and extended OnACID with new initialization methods and a convolutional neural network (CNN), referred to as CaImAn Batch and CaImAn Online, respectively. In general, the accuracy of assumptions in these model-based methods in characterizing the embedded patterns is a critical factor in the performance of such methods (20). For example, CNMF models the background as a low-rank matrix, which might not capture the complex dynamic of the background in one-photon imaging recordings. To compensate for this background, Zhou et al. (16) incorporated an autoregressive model for the background components to process one-photon imaging data.

Deep-learning, or neural networks, can serve as an alternative to the above classic machine learning techniques. CNNs learn hierarchies of informative features for a specific task from labeled datasets (20). Modern fully convolutional neural networks have become a staple for semantic image segmentation, providing an end-to-end solution for the pixel-to-pixel classification problem (21). These networks are often more efficient compared to the traditional CNN-based segmentation approaches that label each pixel of an image based on the local intensity values (21).

A few recent approaches have utilized CNNs to segment neurons from two-dimensional (2D) images for subsequent temporal analysis. These methods treat multiple frames of imaging data as

either additional channels (17) or one image averaged from all frames (the "mean image") (18). One example of this class of CNN-based methods is the method of Apthorpe et al. (17), which applies 2D kernels to individual frames and aggregates temporal information with a temporal max-pooling layer in the higher levels of the network. While the performance was not significantly different from a similar network that only processed the mean image, this CNN method outperformed PCA/ICA. More recently, based on the fully convolutional UNet (22), Klibisz et al. (18) developed the UNet2DS method that segments neurons from the mean image. In general, these methods are suboptimal for differentiating active from non-active neurons due to the loss of temporal dynamics when summarizing temporally collected images into a mean image. Similarly, sparsely firing neurons may appear at unidentifiable contrasts compared to the background after undergoing averaging to the mean image. Lastly, 2D segmentation of mean images has difficulty in delineating the neuron boundaries between overlapping neurons that independently fire in time (Fig. 1).

Three-dimensional (3D) CNN architectures could be superior to 2D segmentation networks as they have the advantage of incorporating temporal information into an end-to-end learning process (23). Compared to methods that process 2D images, spatio-temporal methods can provide more accurate results in identifying sparsely spiking and overlapping neurons, but are also computationally more challenging (13). Compared to iterative methods such as CNMF, a 3D CNN architecture could produce high computational efficiency for long-duration, large-scale recordings. 3D CNNs have already been impactful in other video (23, 24) and volumetric biomedical (25-27) data analyses.

A critical factor prohibiting development and accurate assessment of such novel learning-based techniques (e.g. 3D CNNs) is the absence of a comprehensive public dataset with accurate gold-standard ground truth markings. Indeed, the Allen Brain Observatory (ABO) (http://observatory.brain-map.org/visualcoding) and the Neurofinder challenge (https://github.com/codeneuro/neurofinder) have provided invaluable online resources in the form of diverse datasets spanning multiple brain areas. We demonstrate that existing markings that accompany these datasets contain significant errors, further complicating algorithm development and assessment. Like many other medical imaging modalities that lack empirically-driven ground truth, human expert markings could serve as the gold-standard. In such situations, the agreement between multiple expert human graders has traditionally determined the practical upper bound for

accuracy. No automated algorithm to-date is shown to be closer in accuracy to the markings of an expert human grader than another experienced grader.

In this paper, we present a novel CNN-based method with spatio-temporal convolutional layers to segment active neurons from two-photon calcium imaging data. To train and validate the performance of this algorithm, we utilize online datasets from the ABO and Neurofinder challenge. Since we show that the original manual markings that accompany these datasets are imperfect, we carefully manually-relabel active neurons in these datasets. We compare the performance of our network with other state-of-the-art neuron segmentation methods on these datasets. The results indicate that our trained network is fast, superior to other methods, and achieves human accuracy. To demonstrate the generalizability of our method, we show that the network trained on data recorded at a specific cortical layer from the ABO dataset can also accurately segment active neurons from other layers and cortical regions of the mouse brain with different neuron types and densities. We demonstrate that adding region-specific recordings to the ABO training set significantly improves the performance of our method. To promote future advancement of neuron segmentation algorithms, we provide the manual markings, source code for all developed algorithms, and weights of the trained networks as an open-source software package.

RESULTS

Spatio-temporal neuron segmentation using deep-learning. The key feature of our active neuron segmentation framework (Fig. 2*A*) was a new 3D CNN architecture, which we named Spatio-Temporal NeuroNet (STNeuroNet) (Fig. 2*B* and *SI Appendix*, Fig. S1). The 3D convolutional layers in STNeuroNet extracted local spatio-temporal information that capture the temporal dynamics of the input recording. STNeuroNet consisted of downsampling, upsampling, convolutional skip connections, and temporal max-pooling components that predict neuron masks based on spatio-temporal context of the input recording. The network generated feature maps at three different resolutions with a cascade of dense feature stacks and strided convolutional layers. The network then upsampled and fused the extracted features to generate the final predictions. After initial background compensation of individual movie frames, STNeuroNet processed sequences of short temporal batches of $N = 120$ frames and output a 2D probability map of active neurons for each batch. We then applied an optimal threshold to the neuron probability maps and automatically separated high probability regions into individual neuron instances. Lastly, the final

set of unique active neurons for the entire recording was determined by eliminating duplicate masks of the same neurons that were identified in different temporal intervals of the video (*Methods*).

STNeuroNet accurately segmented neurons from the Allen Brain Observatory dataset. We first quantified the performance of our method using a subset of the ABO dataset. This dataset included the first 12 minutes and 51 seconds of two-photon microscopy recordings from 275 μm deep in the primary visual cortex (VISp) of ten mice expressing the GCaMP6f calcium sensor. We binned these videos from 30 Hz to 6 Hz to speed up the processing time without significantly compromising the neuron identification results (12, 16, 28) and to permit uniform comparison across future datasets with similar imaging rates.

The Allen Institute performed automatic neuron segmentation without manual inspection of the results (29). We inspected the provided set of masks and found that some of the masks did not correspond to active neurons in the selected ~13 minutes time-interval, and some active neurons were not included in the set (*SI Appendix*, Fig. S2). Thus, two expert human graders improved the accuracy by sequentially editing the labeling and creating the gold-standard ground truth (GT) labels (*Methods*; *SI Appendix*, Table S1). Overall, we removed $n = 40 \pm 23.6$ masks (mean ± standard deviation over $n = 10$ videos) from the initial ABO marking as they were not located on the soma of active neurons, accounting for 13.9 ± 5.7% of the initial neurons, and added $n = 72.7 \pm 20.9$ neurons, accounting for 24.2 ± 5.9% of the final GT neurons. The final set of neurons comprising the GT demonstrated peak calcium responses with $d' \geq 4$ within the spike detection formalism (*Methods*), which were at significantly higher levels compared to the distribution of d' values from the baseline due to noise (p-values < 0.001, one-sided Z-test using $n = 500$ baseline samples for each of the 3016 GT neurons). To optimally utilize our labeled dataset yet strictly separate training and testing datasets, we used leave-one-out cross-validation to assess the performance of our algorithm for detection and segmentation of active neurons.

Training our network on 144×144×120 segments of input data took 11.5 hours for 36,000 iterations. After training, STNeuroNet generated neuron predictions in 171.24 ± 21.28 seconds (mean ± standard deviation over $n = 10$ videos) when processing 4624 ± 5 frames of size 487×487 pixels. The complete framework, from preprocessing to the final neuron aggregation, processed these recordings with 17.3 ± 1.2 frames/s (mean ± standard deviation over $n = 10$ videos) speed.

Note that considering the binning of videos from 30 Hz to 6 Hz, the effective processing rate can be up to 5 times better than the reported number.

Fig. 3 shows an illustrative example of our framework applied on a time-interval of $N = 1200$ background compensated frames from one mouse, which achieved neuron detection scores (recall, precision, F_1) of (0.86, 0.88, 0.87) (*Methods*). The first frame, last frame, and the normalized temporal average of all frames in the batch are shown in Fig. 3*B*. To better illustrate temporal neuronal activity, we also show the correlation image, defined as the mean correlation value between each pixel with its 4-connected neighborhood pixels. Temporal $\Delta F/F$ traces (where ΔF is the difference between the signal peak and baseline amplitudes, and F is the mean baseline amplitude) for selected true positive, false negative, and silent neurons highlight the presence or absence of activity in the selected time-interval, indicating that STNeuroNet effectively selected active neurons while disregarding silent neurons (Fig. 3*B*-*C*).

Using the same ten videos, we compared the performance of our framework to the performance of CaImAn Online and CaImAn Batch (19), Suite2p (12), HNCcorr (15), and to the deep-learning based UNet2DS (18) algorithm, quantifying each algorithm in terms of recall, precision, and F_1 (Fig. 4). To compare all algorithms on an equal footing, we optimized the algorithmic parameters for each method through leave-one-out cross-validation (*SI Appendix*, *Supplementary Methods*). Since F_1 quantifies a balance between recall and precision, we used this score as the final metric to optimize and assess the performance of all methods. Our framework outperformed all other algorithms in the F_1 score (p-value < 0.005, two-sided Wilcoxon rank sum test over $n = 10$ videos; Fig. 4*A* and *SI Appendix*, Table S2) at higher speed compared to CaImAn Batch and HNCcorr (p-values < 0.005, two-sided Wilcoxon rank sum test over $n = 10$ videos), while being as fast as CaImAn Online and slower than Suite2p (p-values = 0.3075 and < 0.005, respectively; two-sided Wilcoxon rank sum test over $n = 10$ videos; Fig. 4*B*) when processing 487×487 pixels videos. After disregarding the initialization time of STNeuroNet, our framework was significantly faster than Suite2p (p-values = 0.026, two-sided Wilcoxon rank sum test over $n = 10$ videos). For CaImAn Online, the initialization time was 10.4 ± 0.8 s for 100 frames and did not contribute significantly to the total processing time. Because UNet2DS processed a single 2D image, it was extremely fast (speed = 2263.3 ± 2.6 frames/s for $n = 10$ videos), but it was not able to separate

overlapping neurons, resulting in low recall values compared to other methods (*SI Appendix*, Fig. S3*A*).

We further investigated the underlying source for our framework's superior recall compared to other spatio-temporal methods. Fig. 4*C-D* and *SI Appendix*, Fig. S3*B-E* illustrate examples of sparsely-firing neurons with low $\Delta F/F$ value calcium transients that were identified by STNeuroNet and missed by other algorithms. We further validated this observation by quantifying the percentage of GT neurons detected at different levels of peak signal-to-noise ratio (PSNR; *Methods*) in Fig. 4*E*. STNeuroNet's higher percentage of true positive neurons compared to other algorithms in the low PSNR regime indicates that our network achieved high recall because it identified a larger portion of spiking neurons with relatively low PSNR calcium transients. On average, our algorithm detected 22.4 ± 7.5%, 7.9 ± 3.6%, 21.0 ± 4.8%, 26.1 ± 4.6%, and 38.1 ± 5.9% more neurons (mean ± standard deviation for n = 10 videos) from the GT compared to CaImAn Online, CaImAn Batch, Suite2p, HNCcorr, and UNet2DS, respectively.

To assess the reproducibility of our GT markings, we trained a third grader to conduct an inter-human agreement test. Grader #3 labelled these data from scratch without access to the initial masks from the Allen Institute or the consensus GT segmentations produced by the first two graders. GT and grader #3 were consistent in segmenting neurons with high PSNR (*SI Appendix*, Fig. S4*A-B*). The resulting distribution of mismatched cases (set of missed and falsely-labelled neurons) was weighted towards neurons with low PSNR values, which challenge human perception during manual marking of the video (*SI Appendix*, Fig. S4*B*). Our framework achieved a higher F_1 score compared to grader #3 (mean of 0.84 vs 0.78, p-value = 0.0013; two-sided Wilcoxon rank sum test for n = 10 videos; *SI Appendix*, Fig. S4*C*). To mimic the case of semi-automatic marking, we asked a fourth grader to independently correct the ABO markings for these videos. Compared to grader #4, both grader #3 and STNeuroNet achieved lower F_1 scores (p-values = 0.0002 and 0.0036, respectively; two-sided Wilcoxon rank sum test for n = 10 videos; *SI Appendix*, Fig. S4*C*), which is due to the inherent bias between the GT set and grader #4's markings (*SI Appendix*, Table S3).

The trained STNeuroNet segmented neurons from unseen recordings of additional cortical layers. To demonstrate the generalizability of our trained STNeuroNet, we next applied our segmentation framework to recordings from a different cortical layer in VISp.

We trained STNeuroNet with the same ten videos as in the previous section, from 275 μm below the pia in VISp. The neurons in these datasets were drawn from Rorb-IRES2-Cre mouse line, which restricts expression to layer 4 neurons, and the Cux2-CreERT2 mouse line, which restricts expression to excitatory cell types (*SI Appendix*, Table S4). We then tested this network on data acquired from ten different mice, this time from a different cortical layer at 175 μm deep in VISp. The neurons in these datasets were drawn from the Cux2-CreERT2 and Emx1-IRES-Cre mouse lines, which express calcium sensors in excitatory neurons (*Methods* and *SI Appendix*, Table S4). The data from 175 μm deep is putatively in layer 2/3, while the data from 275 μm deep is at the interface between layer 4 and layer 2/3. Neurons from the test dataset were qualitatively visually different from neurons in the training set (Fig. 5*A*). Quantitatively, the test set had bigger neurons (median of 112.6 μm^2 versus 102.8 μm^2; p-value < 0.005, two-sided Wilcoxon rank sum test over n = 2182 and 3016 neurons, respectively; Fig. 5*B*) and lower densities of identified active neurons (0.0014 ± 0.0002 neurons/μm^2 versus 0.0019 ± 0.0003 neurons/μm^2 for 175 and 275 μm data, respectively; p-value<0.005, two-sided Wilcoxon rank sum test over n = 10 videos). Despite the differences in the size and density of neurons within these two datasets, our network trained on 275 μm data performed at indistinguishable levels on 275 μm test data and 175 μm data (p-value = 0.1212 for F_1; two-sided Wilcoxon rank sum test with n = 10 videos for both groups; Fig. 5*C* and *SI Appendix*, Table S2). Using the layer 275 μm data to set the algorithmic parameters of other methods, our framework achieved the highest mean F_1 score on the 175 μm data (p-value < 0.005, two-sided Wilcoxon rank sum test over n = 10 videos; Fig. 5*D* and *SI Appendix*, Table S2). Unlike our method, the F_1 scores of all other methods except UNet2DS were significantly lower on the 175 μm data compared to the 275 μm test data (p-values = 0.006, 0.031, 0.021, and 0.045 for CaImAn Online, CaImAn Batch, Suite2p, and HNCcorr, respectively; two-sided Wilcoxon rank sum test over n = 10 videos; *SI Appendix*, Table S2).

STNeuroNet accurately segmented neurons from Neurofinder data. We also applied our framework on two-photon calcium imaging data from the Neurofinder challenge. These recordings are from GCaMP6 expressing neurons within different cortical and subcortical regions acquired and labelled by different labs. We used the datasets with activity-informed markings for training and comparison between different algorithms (*Methods*). Upon systematic inspection of Neurofinder GT sets, we found many putative neurons (n = 2, 2, 81, 60, 50 and 19 neurons for datasets called 01.00, 01.01, 02.00, 02.01, 04.00, and 04.01, respectively,

corresponding to 0.5%, 0.6%, 41.1%, 33.7%, 21.2%, and 7.67% of the original GT neurons) with spatial shape and fluorescence temporal waveforms expected from GCaMP6-expressing neurons. Examples of such GT errors from the data called 04.00 in the Neurofinder training set are shown in Fig. 6*A*-*B*. The extracted transients in the time-series of newly-found neurons among all datasets had high detectability index $d' > 3.2$, emphasizing that these signals are truly activity-evoked transients. We also computed the average fluorescence image during these highly detectable transients, which yielded high quality images of the neurons (Fig. 6*B left*).

We analyzed the impact of using different training GT sets on STNeuroNet's performance. The senior grader (grader #1) corrected the labeling of the training data by adding the missing neurons to the GT sets and labelled the Neurofinder test set (*SI Appendix*, Table S1). Compared to the case of using Neurofinder's GT for training, the average F_1 score was not significantly different to the case of employing the markings from grader #1 for both training and testing (p-value = 0.9372, two-sided Wilcoxon rank sum test over n = 6 videos; Fig. 6*C*). Similar to the ABO dataset, we conducted an inter-human agreement test. Independent from grader #1, grader #2 created a second set of markings for the test datasets (*SI Appendix*, Table S1). When tested on grader #1's markings, our algorithm attained comparable average F_1 score to grader #2 (p-value = 0.2403 and 0.3095 for training on Neurofinder's GT and grader #1's GT, respectively; two-sided Wilcoxon rank sum test over n = 6 videos; Fig. 6*C*).

Using our expert manual markings as GT for the Neurofinder dataset, we compared our framework to other methods (Fig. 6*D* and *SI Appendix*, Table S2). For all algorithms, we used the entire Neurofinder training set to optimize the algorithmic parameters for each method (*SI Appendix*, *Supplementary Methods*). Our framework (STNeuroNet trained with the entire training set) achieved higher but statistically insignificant F_1 score than Suite2p (mean ± standard deviation of 0.70±0.03 and 0.61±0.08, respectively; p-value = 0.0649, two-sided Wilcoxon rank sum test over n = 6 videos). Compared to all other methods, STNeuroNet's F_1 score was significantly higher (p-values < 0.005, two-sided Wilcoxon rank sum test over n = 6 videos).

To further test the generalizability of our framework to experimentally-different data, we compared the performance of STNeuroNet trained on the ABO Layer 275 μm dataset to STNeuroNet trained on all Neurofinder training set, when evaluated on the Neurofinder test data (*SI Appendix*, Table S5). Although using the ABO Layer 275 μm data for training resulted in lower

mean F_1 score, the scores were not statistically different (p-value = 0.485, two-sided Wilcoxon rank sum test for n = 6 videos), and the performance was comparable to that of Suite2p (p-value = 1, two-sided Wilcoxon rank sum test for n = 6 videos). With the addition of the high-quality ABO Layer 275 μm data to the Neurofinder training set, STNeuroNet achieved higher F_1 score compared to the network trained only on the Neurofinder training set (p-value = 0.026, two-sided Wilcoxon rank sum test for n = 6 videos; *SI Appendix*, Table S5).

DISCUSSION

In this paper, we presented an automated, fast, and reliable active neuron segmentation method to overcome a critical bottleneck in the analysis workflow of utilizing neuronal signals in real-time behavioral studies. The core component of our method was an efficient 3D CNN named STNeuroNet. The performance of this core was further improved by intuitive pre- and post-processing steps. Our proposed framework for sequential processing of the entire video accurately segmented overlapping active neurons. In the ABO dataset, our method surpassed the performance of CaImAn, Suite2p, HNCcorr, UNet2DS, and an expert grader, and generalized to segmenting active neurons from different cortical layers and regions with different experimental setups. We also achieved the highest mean F_1 score on the diverse datasets from the Neurofinder challenge.

STNeuroNet is an extension of DenseVNet (30), which consists of 3D convolutional layers, to segment active neurons from two-photon calcium imaging data. The added temporal max-pooling layer to the output of DenseVNet summarized the spatio-temporal features into spatial features. This step greatly increased the speed of training and inference processes, which is important for high-speed network validation and low-latency inference in time-sensitive applications such as closed-loop experiments.

We showed the superior performance of our method for active neuron detection and segmentation by direct comparison to the state-of-the-art classic machine learning as well as deep-learning methods. We achieved this level of performance by consistently detecting larger number of true active neurons compared to other algorithms. Our superior performance was not dependent on the GT created by graders #1 and #2 (*SI Appendix*, *Supplementary Experiment*). This is in part due to the fact that unlike the model-based spatio-temporal deconvolution methods of CaImAn and Suite2p, our proposed STNeuroNet extracts relevant spatio-temporal features from the imaging

data without prior modeling; the deep-learning approach could be more flexible for detecting arbitrary spatio-temporal features. Compared to the deep-learning based UNet2DS that is applied to a single aggregate (mean) image, our proposed framework was more powerful in discriminating overlapping neurons and identifying neurons with low activity-evoked contrast because it assesses information in each video frame individually, and in concert with other frames.

One advantage of deep-learning based methods is that once trained, they are computationally fast at inference time. We showed that our framework achieved significantly higher detection scores compared to all other methods at practically high processing speed. While we measured the computational speed of all algorithms on the same computer, we acknowledge that some of these algorithms could potentially benefit from more computationally optimal coding that target other specific hardware architectures. Combined with signal separation (11, 31, 32) and fast spike detection algorithms (32-34), our framework could potentially enable fast and accurate assessment of neural activity from two-photon calcium imaging data. Our current implementation performed neuron detection at near video-rate processing of individual frames when processing sets of sequential frames, which suggests that our framework can interleave updates of the segmentation results with data acquisition. Because our framework can be applied to overlapping or non-overlapping temporal batches, it presents a flexible trade-off to either increase speed or accuracy: processing non-overlapping temporal batches speeds up the algorithm, while using the median or mean probability map of highly overlapping batches could potentially improve the performance at inference time.

Depending on the complexity of the problem and the architecture of neural networks, deep-learning methods need different amount of training data to achieve high performance scores and to be generalizable. We utilized data augmentation, dropout (35), and batch-normalization (36) to achieve generalizability and prevent overfitting. We demonstrated the generalizability of our trained STNeuroNet by applying the processing framework on recordings from different cortical layers and regions (*SI Appendix*, Table S5). We were able to train STNeuroNet on neurons from 275 μm deep in the mouse cortex and segment active neurons from 175 μm deep at an indistinguishable performance level, despite the differences in the neuron size and densities at these two depths. This experiment confirmed that our network was not over-trained to segment active neurons from a specific cortical depth. Adding ABO Layer 275 μm data to the Neurofinder

training dataset improved accuracy of segmenting the Neurofinder test dataset (*SI Appendix*, Table S5). These results suggest that utilizing training data acquired with different experimental setups is beneficial for generalizing STNeuroNet. Also, training on the entire ABO dataset and testing on Neurofinder recordings shows that having more training data from one experimental set up improves performance of segmenting videos from a different experimental set up (*SI Appendix*, Table S5). These experiments confirm that other neuroscientists with significantly different recordings can take advantage of our trained network through transfer learning (37) to adapt the network to their specific data. Combined with transfer learning, our trained network has the potential to achieve high performance and generalizability on experimentally diverse recordings.

In this work, we carefully relabeled active neurons from the ABO dataset to compare the performance of different algorithms. To minimize the probability of human error in marking active neurons, we created the final set of GT masks by combining the markings from two independent graders. To assess human grading consistency, we compared the markings of a third independent grader performing manual segmentation from scratch to the GT. We showed that our framework's performance was higher than grader #3's, suggesting that STNeuroNet learned informative features and surpassed human-level accuracy in active neuron segmentation. For the sake of completeness, we added an additional experiment to reflect the effect of bias in performance of human graders. We compared our method to grader #4, a grader who corrected the ABO dataset markings with similar procedures to, but independently of, graders #1 and #2. As expected, due to the bias created by having access to pilot segmentation labels, grader #4's markings were closer to the GT than grader #3's markings.

Naturally, using manual labeling as the gold-standard has the disadvantage of introducing human errors and bias in the GT data. However, currently available alternative approaches are even less suitable for generating GT. For example, simultaneous dual channel imaging of activity-independent nuclear tagged neurons provides reliable ground truth markings for all neurons. However, such labels which include both active and inactive neurons are not suitable for evaluating segmentation methods for active neurons in behavioral experimentations. Progress in activity-based neuron labeling methods combined with simultaneous optical and structural imaging techniques may provide reliable gold-standard datasets in future.

In addition to the ABO dataset, we also included the results of segmenting the diverse Neurofinder challenge datasets. We included these results because the Neurofinder dataset has been used to assess the accuracy of many recent segmentation algorithms (12, 14, 15, 18). Our framework significantly outperformed all other methods except Suite2p, which could be due to the small sample size and the relatively-large spread of Suite2p's F_1 scores. It is encouraging that our method achieved the highest mean F_1, but our finding that the GT labeling of the training dataset from the challenge has missed neurons is nearly as important. While we do not have access to the labeling of the test dataset, we presume that GT accuracies in the publicly-available training datasets match that of the test data. Thus, we carefully manually labeled the test set in the Neurofinder challenge. The availability of these carefully labeled GT training and test sets are expected to improve the fairness and accuracy of the evaluation metrics to be used for assessing future segmentation algorithms. Similar to the ABO dataset, we achieved above-human-level performance when training on our carefully-labeled markings. Furthermore, when using our carefully curated test labels to evaluate the performance of STNeuroNet under different training conditions, we found that training on our carefully curated training labels only marginally improved performance when compared to training on Neurofinder's labels. This might be due to the nature of the CNN architecture. The architecture seeks to establish a complex yet consistent pattern in data, and could average out erroneous labeling of a subset of the training set as outliers. However, errors in labeling of the test set more affect the performance metrics, as experimentalists use these erroneous labels to directly evaluate the network's output. The impact of training with noisy or incorrect labels on the performance of CNNs is still the subject of active research (38-40), and an in-depth analysis of their effect is beyond the scope of this paper.

We also note that regardless of correct labeling, the limited number of training samples per dataset in the Neurofinder challenge is a major bottleneck for optimal training of CNN-based methods. Our method achieved generalizability and human-level performance, and thus, could assist in the creation of additional accurate training sets for future algorithm development. CNN-generated GT datasets could potentially reduce the workload of human graders while improving the accuracy of the markings by minimizing human errors due to subjective heuristics.

This work is the first step in a continuum of research to utilize 3D CNNs for detection and segmentation of neurons from calcium imaging data. The data used in our work were properly

corrected for motion artifacts by the data owners. In the more general case of non-registered datasets, algorithms such as NoRMCorre (41) can be used to accurately correct motion prior to the application of our framework. We used watershed to separate the identified overlapping neurons co-activated in the same time interval processed by STNeuroNet, which can give inaccurate masks. Since such overlapping neurons might segregate themselves in other time intervals, we presented the neuron fusion process to circumvent this issue and obtain masks that had overlapping pixels. Each component of our method, individually or together, can be used by us and other researchers in many related projects. To this end, as our computationally fast and accurate method is an invaluable tool for a large spectrum of real-time optogenetic experiments, we have made our open-source software and carefully annotated datasets freely available online. Future work will extend the current framework to minimize parameter adjustments in pre- and post-processing steps by encapsulating these steps into an end-to-end learning process. Such an approach would remove the need for watershed-based separation of overlapping neurons, which is prone to error for one-photon recordings or two-photon imaging of species or brain areas with significantly overlapping populations, which was not present in the data utilized in our work.

METHODS

Proposed active neuron segmentation method. Fig. 2 outlines the proposed segmentation algorithm, which contains three major components. First is a set of preprocessing steps to make two-photon microscopy data appropriate for analysis by CNN. Second is our core 3D CNN architecture, named STNeuroNet, that generates a probability map of potential masks for active neurons from these preprocessed data. Third, and final stage is a set of post-processing steps to infer the location and mask of individual active neurons from the outputs of STNeuroNet. These steps are discussed in detail in the following sections.

Image preprocessing steps. All data used in our work were previously registered. We first cropped the boundary region of the data to remove black borders introduced in the registration processes (10 μm in each direction for the ABO data and 4-50 μm for the Neurofinder data). To increase SNR, reduce the computational complexity, and allow utilization of the trained network for future data with different recording speeds, we temporally binned ABO and Neurofinder videos to 6 Hz and 3 Hz videos (lowest frame rate among the five datasets in the Neurofinder challenge),

respectively. We performed temporal binning by combining a set of consecutive frames into one frame via summation. We then corrected for non-uniform background illumination using homomorphic filtering (42) on each frame of the video. We formulated a high-pass filter by subtracting a low-pass Gaussian filter with standard deviation of 0.04 μm^{-1} from 1. Then, we normalized the intensity of each video by dividing by its overall standard deviation.

Neural network architecture. Much like action recognition from videos, active neuron segmentation requires capturing context from multiple frames. This motivated us to utilize 3D convolutional layers in our deep-learning network. 3D convolutional layers extract local spatio-temporal information that capture the temporal dynamics of the input recording. We used the DenseVNet (30), implemented as part of NiftyNet (43), as the backbone for our STNeuroNet network. Like other popular fully CNNs for semantic segmentation of medical images (e.g. UNet (22) and VNet (27)), DenseVNet is composed of downsampling (or encoder), upsampling, and skip connection components (Fig. 2*B* and *SI Appendix,* Fig. S1). Unlike the two previous networks, each encoder stage of DenseVNet is a dense feature stack. The input to each convolutional layer of the stack is the concatenated outputs from all preceding layers of the stack. This structure has the main advantage of improved performance with substantially fewer parameters through gradient propagation and feature reuse (30, 44). In the encoder path, strided convolutional layers reduce the dimensionality of the input feature map and connect dense feature stacks. Single convolutional layers in the skip connections, followed by bilinear upsampling, transform the output feature maps from each stage of the encoder path to the original image size (30). All convolutional layers in DenseVNet perform 3D convolutions, use the rectified linear unit (ReLU) non-linearity as the activation function, and consist of batch normalization (36) and dropout (35) with probability of 0.5 (except the last layer). Unlike the original implementation of the network, we did not use spatial priors, dilated convolutions, and batch-wise spatial dropout, as these did not have a significant effect on the final results reported in the original paper (30).

We made the following two modifications to DenseVNet for our application: (1) we changed the last convolutional layer of DenseVNet to have ten output channels instead of the number of classes, and (2) we added a temporal max-pooling layer to the upsampled features, followed by a 2D convolutional layer with ten 3×3 kernels, and a final convolutional layer with two 3×3 kernels to the output of DenseVNet. The temporal max-pooling layer summarizes the extracted temporal

feature maps, greatly increasing the speed of the training process and reducing inference time by reducing the number of output predictions (2D predictions instead of 3D predictions). This step is important for high-speed network validation and low-latency inference. The last convolutional layer computes two feature maps for the background and neuron classes. We applied Softmax to each pixel of the final feature maps to transform them into probability maps. We used the Dice-loss objective function (27) during training, defined as

$$\text{Dice} - \text{loss} = 1 - \frac{2\sum_{i=1}^{N} p_i q_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} q_i^2}, \quad [1]$$

where the summation is over *N*, the total number of pixels, and p_i and q_i are the Softmax output and GT label for pixel *i*, respectively. The Dice-loss is suitable for binary segmentation problems and handles highly unbalanced classes without the need for sample re-weighting (27).

Training procedure and data augmentation. To create a large set of training samples, we cropped smaller windows of size 144×144×120 voxels from the rotated (0°, 90° and 180°) training videos and GT markings and applied random flips during training. We performed cropping using a spatio-temporal sliding window process with 75% overlap between adjacent windows. Within this large set of samples, we kept samples that contained at least one active neuron in the selected 120 frames time-interval. We trained the networks using sample-level whitening, defined as

$$\frac{I - \text{mean}(I)}{\text{std}(I)}, \quad [2]$$

where *I* is the 3D input sample to the network. We used the Adam optimizer (45) with learning rate of 0.0005 and mini-batch size 3. We trained the ABO and Neurofinder networks for at least 35,000 iterations, or until the loss function converged (maximum 40,000 iterations).

Post-processing steps. *Binarizing Neuron Probability Maps.* We used the entire spatial extent of video frames at test time to estimate the neuron probability maps, which we processed to isolate individual neurons. We processed video frames in non-overlapping batches of $N = 120$ frames, equal to the number of frames used during training. We binarized the probability maps by applying the optimal threshold that yielded the highest mean F_1 score on the training set (*SI Appendix*, Fig. S5). We then separated potential overlapping active neurons from each binarized map and removed small regions. Finally, we aggregated all identified active neuron masks from different time-

intervals to obtain the segmentation masks for the entire recording. These steps are described in detail in the following subsections.

Instance Segmentation. The temporal max-pooling layer in our network merges overlapping active neurons in the segmentation mask. To separate these neurons, we used the watershed algorithm (46). We first calculated the distance transform image as the distance of each pixel to the nearest background pixel. We then applied the MATLAB (Mathworks, Natick, MA) *watershed* function to the distance transform of connected components which had an area greater than a predefined threshold, empirically set to the average neuron area (107.5 μm^2 for ABO and 100-200 μm^2 for Neurofinder). After separating neuron instances, we discarded small segmented regions as background, with the minimum area determined to maximize the mean F_1 score across the training set (*SI Appendix*, Fig. S5). Since the watershed algorithm alone cannot accurately determine neuron boundaries for overlapping cases, we used segmentation results from multiple temporal batches to yield the final neuron masks. This step is detailed in the following section.

Neuron Fusion. Since STNeuroNet outputs a single 2D probability map of active neurons for the input time-interval, we processed two-photon video recordings in subsequent short temporal intervals to better resolve overlapping neurons. Unlike the approach of (17) which used the network predictions to find neuron locations, we used STNeuroNet's predictions to determine the final neuron masks. In each of these time-intervals, we identified and segmented active neurons. Because neurons may activate independently and spike in different times, we aggregated the segmentation results from all time-intervals to attain the segmentation for the entire recording. Aggregation of neuron masks from multiple inferences corresponding to different time-intervals was done in two steps. First, we matched neurons between these segmentations to identify if the same neuron was segmented multiple times and kept the mask with the mean size. We used the distance between the masks' center of mass for this step. Masks with distance smaller than 4 μm were identified as the same neuron. Second, we removed any mask that encompassed one or more neurons from other time-intervals. We removed any mask m_i that overlapped with mask m_j such that

$$\text{Normalized Overlap}(m_i, m_j) = \frac{|m_i \cap m_j|}{|m_j|} > \theta_p, \quad [3]$$

where θ_p is the overlap threshold, which we empirically set to 0.75.

Allen Brain Observatory dataset and labeling. This dataset consists of two-photon recordings from neurons across different layers and areas of mouse visual cortex. Transgenic Cre-line mice drove expression of the genetically encoded calcium indicator GCaMP6f. *SI Appendix,* Table S4 shows the correspondence between the mouse lines and videos used in this paper. We used recordings at 275 μm deep in the cortex of ten mice for comparison between algorithms and recordings at 175 μm deep in the cortex from a set of ten different mice to assess the generalizability of all method.

The data was previously corrected for motion and had an accompanying set of automatically identified neurons (29). We used these automatically detected neurons as initializations for our manual labeling. We developed a custom software with graphical user interface (GUI) in MATLAB 2017b (Mathworks, Natick, MA) that allowed our graders to add to the initial set by drawing along the boundary of newly found neurons (phase 1) and to dismiss wrongly segmented neurons that do not correspond to the soma of an active neuron (phase 2). In phase 1, the GUI provided simultaneous visualization of the video overlaid with segmented neurons' masks on two separate panels. On one panel, background corrected video and in the other panel a summary image of choice (mean, max-projected, or correlation image, defined as the mean correlation value between each pixel with its 4-connected neighborhood pixels) were displayed. In phase 2, the GUI showed the zoomed-in region of the video for each segmented neuron in three panels, which included the background corrected video, the mean image, and the $\Delta F/F$ trace of the average pixel intensities within the neuron's mask. Graders used the following criteria to label each marked mask as neuron: 1) the marked area had a bright ring with a dark center that changed brightness during the recording, or 2) the area was circular and had a size expected from a neuron (10-20 μm in diameter) that changed brightness during the recording. Criterion 1 filters for nuclear-exported protein calcium sensors used by the ABO investigators, while criterion 2 filters for spatio-temporal features of neuron somas that have calcium activity transients.

Two graders independently corrected the markings of the ABO dataset. Matching marks from the two graders were labeled as true neurons, whereas disagreements were reevaluated by the senior grader (grader #1). This grader, blind to the identity of the non-matching masks (meaning who marked it), used the phase 2 of our GUI to assess all disagreements and label the masks as neuron or not a neuron. The set of masks marked by both graders and the set of masks that corresponded

to active neurons from the disagreement set comprised the final GT masks. We created spatio-temporal neuron labels for training by extracting the neurons' active time-intervals. We first separated traces of overlapping neurons using the linear regression approach of (29). Using the extracted time-series for each neuron, we removed neuropil signal, scaled by factor of 0.7 (1), and removed any remaining background fluctuation using a 60 s moving-median filter. For each neuron mask, we defined the neuropil signal as the average fluorescence value in an annulus of 5 μm around the neuron mask, from which we excluded pixels that belonged to other neurons. We found activity-evoked calcium transients with high detection fidelity using tools from statistical detection theory (*Spike detection and discriminability index* section). We considered neurons as active until 0.5 seconds after the detected spike times, equal to 3.5 times the half-decay time of spike-evoked GCaMP6f fluorescence signals reported in (1).

Neurofinder dataset and labeling. The Neurofinder challenge consists of nineteen training and nine testing two-photon calcium imaging datasets acquired and annotated by four different labs. These datasets are diverse: they reported activity from different cortical and subcortical brain regions and varied in imaging conditions such as excitation power and frame rate. The GT labels were available for the training sets, while they were held out for the test set.

The first dataset (called the 00 set) segmented neurons using fluorescently labeled anatomical markers, while others were either manually-marked or curated with a semi-automatic method. Upon inspection of the fourth dataset (called the 03 set), we found that this dataset was labelled based on anatomical factors. We excluded the first and fourth sets from the comparison in the Results section because these datasets would include silent neurons; the activity-independent marking is incompatible for assessing active neuron segmentation methods. The remaining datasets referred to as 01, 02, and 04 each had two training videos. Similar to ABO, we created spatio-temporal labels for the Neurofinder training set by detecting neuronal spikes that satisfied the minimum required d' (*Spike detection and discriminability index* section), which we iteratively reduced down to $d' = 0.5$ if a spike was not identified.

Spike detection and discriminability index. Using tools from statistical detection theory (47, 48), we detected prominent spike-evoked fluorescence signals and quantified their detection fidelity. Specifically, we performed a matched filter approach with an exponentially decaying

signal as the template (*S*), with mean decay time of τ, on the $\Delta F/F$ traces to reduce the effect of noise on spike detection (48):

$$L = F_0 \sum_{i=1}^{n}[-S_i + (1 + (\Delta F/F)_i)\ln(1 + S_i))], \quad [4]$$

in which the summation is over a sliding window of length *n*, and F_0 is the baseline photon-rate. Using the relationship between the mean decay time τ and half-decay time $\tau_{1/2}$ as

$$\tau = \frac{\tau_{1/2}}{\ln(2)}, \quad [5]$$

we used 0.8 s and 0.2 s as the value of τ for GCaMP6s and GCaMP6f data in *S*, respectively. We detected spikes as local-maxima time points in a 1 s window of the filtered signal (*L*) that passed a predefined threshold of γ:

$$\gamma = \mu + \sigma\Phi^{-1}(P_N), \quad [6]$$

which was determined by the tolerable probability of false-negative (P_N) and the mean (μ) and standard deviation (σ) of the distribution of *L* under the hypothesis of a spike having occurred (48). In the above equation, $\Phi^{-1}(.)$ is the inverse of the standard Gaussian cumulative distribution function (48).

We further narrowed down the true spikes using the discriminability index, d', which characterizes the detection fidelity by considering the amplitude and temporal dynamics of the fluorescence signals (48). Higher values of d'provide higher spike detection probabilities and lower errors, with $d' \geq 3$ achieving area under the receiver operating characteristic curve (a metric for spike detectability) greater than 0.98 (48). We determined the minimum required detectability index (d'_{min}) for labeling spikes with the aim of balancing the number of false-positive (P_F) and false-negative (P_N) errors (47):

$$(f_s - \lambda)P_F = \lambda P_N, \quad [7]$$

$$d'_{min} = \Phi^{-1}(1 - P_N) - \Phi^{-1}(P_F) = \Phi^{-1}(1 - P_N) - \Phi^{-1}(P_N\lambda(f_s - \lambda)^{-1}). \quad [8]$$

In Equation [7], f_s and λ denote the recording and neuron spike rates, respectively. For the ABO dataset, since the majority of mice were stationary during the visual stimulus behavior (*SI Appendix*, Fig. S6), we selected λ = 2.9 spikes/s in accordance to previous experimentally obtained spike rates during similar behaviors (49). We then set a low P_N = 0.035, which corresponded to a spike detection threshold of *d'* = 3.6 based on Equations [7-8]. For the Neurofinder challenge, we

used a lower threshold of *d'* = 1.7 to compensate for the overall lower SNR of the data compared to the ABO dataset.

Quantification of peak signal-to-noise ratio (PSNR). To calculate the PSNR of neurons, we first separated traces of overlapping neurons using the linear regression approach of (29). We then removed neuropil signal, scaled by factor of 0.7 (1), and removed any remaining background fluctuation using a 60 s moving-median filter. We then calculated the PSNR for neural traces as

$$\mathrm{PSNR} = \frac{\Delta F_{peak}}{\sigma_n}, \quad [9]$$

where ΔF_{peak} is the difference between the biggest spike value and the baseline value, and σ_n is the noise standard deviation calculated from non-active intervals of traces.

Evaluation metrics. We evaluated segmentation methods by comparing their results with the manual GT labels. We assessed each algorithm by quantifying three neuron detection metrics: recall, precision, and F_1 score, defined as follows:

$$\mathrm{Recall} = \frac{N_{TP}}{N_{GT}}, \quad [10]$$

$$\mathrm{Precision} = \frac{N_{TP}}{N_{detected}}, \quad [11]$$

$$F_1 = 2\frac{\mathrm{Recall}\times\mathrm{Precision}}{\mathrm{Recall}+\mathrm{Precision}}. \quad [12]$$

These quantities derive from the number of manually labelled neurons (ground truth neurons, N_{GT}), number of detected neurons by the method ($N_{detected}$), and number of true positive neurons (N_{TP}). We used the Intersection-over-Union (IoU) metric along with the Hungarian algorithm to match masks between the GT labels and the detected masks (19). The IoU for two binary masks, m_1 and m_2, is defined as

$$\mathrm{IoU}(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}. \quad [13]$$

We calculated the distance between any pair of masks from the GT (m_i^{GT}) and the detected set (M_j) as described by (19):

$$\mathrm{Dist}\left(m_i^{GT}, M_j\right) = \begin{cases} 1 - \mathrm{IoU}\left(m_i^{GT}, M_j\right), & \mathrm{IoU}\left(m_i^{GT}, M_j\right) \geq 0.5 \\ 0, & m_i^{GT} \subseteq M_j \text{ or } M_j \subseteq m_i^{GT} \\ \infty, & \text{otherwise.} \end{cases} \quad [14]$$

In the above equation, a distance of infinity denotes masks that are not matching due to their small IoU score. Next, we applied the Hungarian algorithm to solve the matching problem using the distance matrix defined in Equation [14], yielding the set of true positive masks.

Speed analysis. For each algorithm, we calculated the speed by dividing the number of frames by the processing time (excluding read and write times). For CaImAn Batch, we used all of the logical Cores of our CPU (28 threads) for parallel processing. For STNeuroNet and CaImAn online, we calculated an initialization-independent speed by disregarding the algorithms' initialization times, which were the prefetching of the first batch and the initialization of the components, respectively.

Hardware used. We ran CaImAn, Suite2p, HNCcorr, and the pre- and post-processing part of our algorithm on a Windows 10 computer with Intel Xeon E5-2680 v4 CPU and 256 GB RAM. We trained and tested STNeuroNet and UNet2DS using a single NVIDIA GeForce GTX Titan X GPU. All CNNs in the CaImAn package were deployed on the NVIDIA GeForce GTX Titan X GPU.

Quantification and statistical analysis. Statistical parameters including the definitions and exact values of n (number of frames, number of videos, or number of neurons), location and deviation measures are reported in the Figure Legends and corresponding sections in the main text. All data were expressed as mean ± standard deviation. We used two-sided Z-test for the statistical analysis of calcium transients' d' compared to the distribution of d' values from the baseline due to noise. For all other statistical tests, we performed two-sided Wilcoxon rank sum test; n.s.: not significant, *: p-value < 0.05, and **: p-value < 0.005. We determined results to be statistically significant when p-value < 0.05. We did not remove any data from statistical analyses as outliers.

Data and software availability. Codes for STNeuroNet and all other steps in our algorithm, along with the trained network weights and manual markings are available online in our GitHub repository (https://github.com/soltanianzadeh/STNeuroNet).

ACKNOWLEDGMENTS

We thank David Feng and Jerome Lecoq from the Allen Institute for providing the ABO data, Saskia de Vries and David Feng from the Allen Institute for useful discussions, Hao Zhao for the initial implementation of the GUI, and Leon Kwark for the manual marking of the data. This work

was funded in part by the National Institutes of Health Medical Imaging Training Program pre-doctoral fellowship (T32-EB001040) and P30-EY005722, and the National Science Foundation BRAIN Initiative (NCS-FO 1533598). Y.G is also supported by the Beckman Young Investigator Award. S.F. is also supported by a Google Faculty Research Award.

REFERENCES

1. Chen T-W*, et al.* (2013) Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* 499(7458):295-300.
2. Dombeck DA, Harvey CD, Tian L, Looger LL, & Tank DW (2010) Functional imaging of hippocampal place cells at cellular resolution during virtual navigation. *Nature neuroscience* 13(11):1433-1440.
3. Ghosh KK*, et al.* (2011) Miniaturized integration of a fluorescence microscope. *Nature methods* 8(10):871-878.
4. Grienberger C & Konnerth A (2012) Imaging calcium in neurons. *Neuron* 73(5):862-885.
5. Yang W & Yuste R (2017) In vivo imaging of neural activity. *Nature methods* 14(4):349-359.
6. Andilla FD & Hamprecht FA (2014) Sparse space-time deconvolution for calcium image analysis. *Advances in Neural Information Processing Systems*, pp 64-72.
7. Giovannucci A*, et al.* (2017) OnACID: Online Analysis of Calcium Imaging Data in Real Time. *Advances in Neural Information Processing Systems*, pp 2378-2388.
8. Guan J*, et al.* (2018) NeuroSeg: automated cell detection and segmentation for in vivo two-photon Ca2+ imaging data. *Brain Structure and Function* 223(1):519-533.
9. Kaifosh P, Zaremba JD, Danielson NB, & Losonczy A (2014) SIMA: Python software for analysis of dynamic fluorescence imaging data. *Frontiers in neuroinformatics* 8:80.
10. Maruyama R*, et al.* (2014) Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Networks* 55:11-19.
11. Mukamel EA, Nimmerjahn A, & Schnitzer MJ (2009) Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron* 63(6):747-760.
12. Pachitariu M*, et al.* (2017) Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *BioRxiv*:061507.
13. Pnevmatikakis EA*, et al.* (2016) Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron* 89(2):285-299.
14. Reynolds S*, et al.* (2017) ABLE: An Activity-Based Level Set Segmentation Algorithm for Two-Photon Calcium Imaging Data. *eNeuro* 4(5):ENEURO. 0012-0017.2017.
15. Spaen Q, Hochbaum DS, & Asín-Achá R (2017) HNCcorr: A Novel Combinatorial Approach for Cell Identification in Calcium-Imaging Movies. *arXiv preprint arXiv:1703.01999*.
16. Zhou P*, et al.* (2018) Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *eLife* 7:e28728.
17. Apthorpe N*, et al.* (2016) Automatic neuron detection in calcium imaging data using convolutional networks. *Advances in Neural Information Processing Systems*, pp 3270-3278.
18. Klibisz A, Rose D, Eicholtz M, Blundon J, & Zakharenko S (2017) Fast, Simple Calcium Imaging Segmentation with Fully Convolutional Networks. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, (Springer), pp 285-293.
19. Giovannucci A*, et al.* (2019) CaImAn: An open source tool for scalable Calcium Imaging data Analysis. *eLife* 8:e38173.

20. LeCun Y, Bengio Y, & Hinton G (2015) Deep learning. *Nature* 521:436-444.
21. Long J, Shelhamer E, & Darrell T (2015) Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3431-3440.
22. Ronneberger O, Fischer P, & Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer), pp 234-241.
23. Tran D, Bourdev L, Fergus R, Torresani L, & Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. *Computer Vision (ICCV), 2015 IEEE International Conference on*, (IEEE), pp 4489-4497.
24. Varol G, Laptev I, & Schmid C (2017) Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence* 40(6):1510-1517.
25. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, & Ronneberger O (2016) 3D U-Net: learning dense volumetric segmentation from sparse annotation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer), pp 424-432.
26. Kamnitsas K*, et al.* (2017) Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis* 36:61-78.
27. Milletari F, Navab N, & Ahmadi S-A (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. *3D Vision (3DV), 2016 Fourth International Conference on*, (IEEE), pp 565-571.
28. Friedrich J*, et al.* (2017) Multi-scale approaches for high-speed imaging and analysis of large neural populations. *PLoS computational biology* 13(8):e1005685.
29. de Vries SE*, et al.* (2018) A large-scale, standardized physiological survey reveals higher order coding throughout the mouse visual cortex. *bioRxiv*:359513.
30. Gibson E*, et al.* (2018) Automatic multi-organ segmentation on abdominal CT with dense v-networks. *IEEE Transactions on Medical Imaging* 37(8):1822-1834.
31. Bell AJ & Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. *Neural computation* 7(6):1129-1159.
32. Vogelstein JT*, et al.* (2010) Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of neurophysiology* 104(6):3691-3704.
33. Deneux T*, et al.* (2016) Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature communications* 7:12190.
34. Friedrich J, Zhou P, & Paninski L (2017) Fast online deconvolution of calcium imaging data. *PLoS computational biology* 13(3):e1005423.
35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, & Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929-1958.
36. Ioffe S & Szegedy C (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, pp 448-456.
37. Yosinski J, Clune J, Bengio Y, & Lipson H (2014) How transferable are features in deep neural networks? *Advances in neural information processing systems*, pp 3320-3328.
38. Rolnick D, Veit A, Belongie S, & Shavit N (2017) Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.
39. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, & Fergus R (2014) Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.
40. Zhang C, Bengio S, Hardt M, Recht B, & Vinyals O (2016) Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.
41. Pnevmatikakis EA & Giovannucci A (2017) NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of neuroscience methods* 291:83-94.

42. Oppenheim Av, Schafer R, & Stockham T (1968) Nonlinear filtering of multiplied and convolved signals. *IEEE transactions on audio and electroacoustics* 16(3):437-466.
43. Gibson E*, et al.* (2018) NiftyNet: a deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine* 158:113-122.
44. Huang G, Liu Z, Weinberger KQ, & van der Maaten L (2017) Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p 3.
45. Kingma DP & Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
46. Meyer F (1994) Topographic distance and watershed lines. *Signal processing* 38(1):113-125.
47. Soltanian-Zadeh S, Gong Y, & Farsiu S (2018) Information-Theoretic Approach and Fundamental Limits of Resolving Two Closely-Timed Neuronal Spikes in Mouse Brain Calcium Imaging. *IEEE Transactions on Biomedical Engineering* 65(11):2428-2439.
48. Wilt BA, Fitzgerald JE, & Schnitzer MJ (2013) Photon shot noise limits on optical detection of neuronal spikes and estimation of spike timing. *Biophysical journal* 104(1):51-62.
49. Niell CM & Stryker MP (2010) Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron* 65(4):472-479.

FIGURE LEGENDS

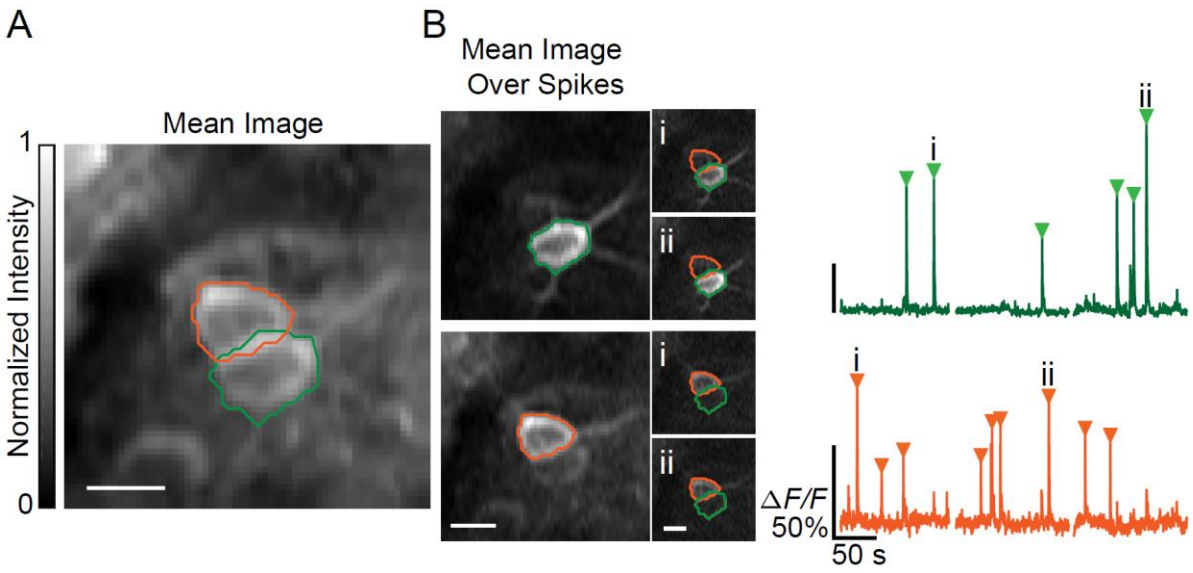


Fig. 1. Overlapping neurons complicate active neuron segmentation. (*A*) Neurons can have overlapping regions in two-photon calcium imaging data due to the projection of a 3D volume onto a 2D imaging plane, as evident in the mean image, normalized to the maximum intensity of the cropped region. (*B*) The temporal evolution of neuron intensities provides important information for accurate segmentation of such cases, which is exploited by the method proposed in this paper. The time-series in green and orange correspond to neurons outlined with matching colors. Images in the middle panel show the recorded data at the marked time-points, and the images in the left panel are the normalized mean images of frames corresponding to each neuron's active time-interval (defined as 0.5 seconds after the marked spike times). We separated traces of these overlapping neurons using the linear regression approach of the Allen Institute (29). Scale bars are 10 µm.

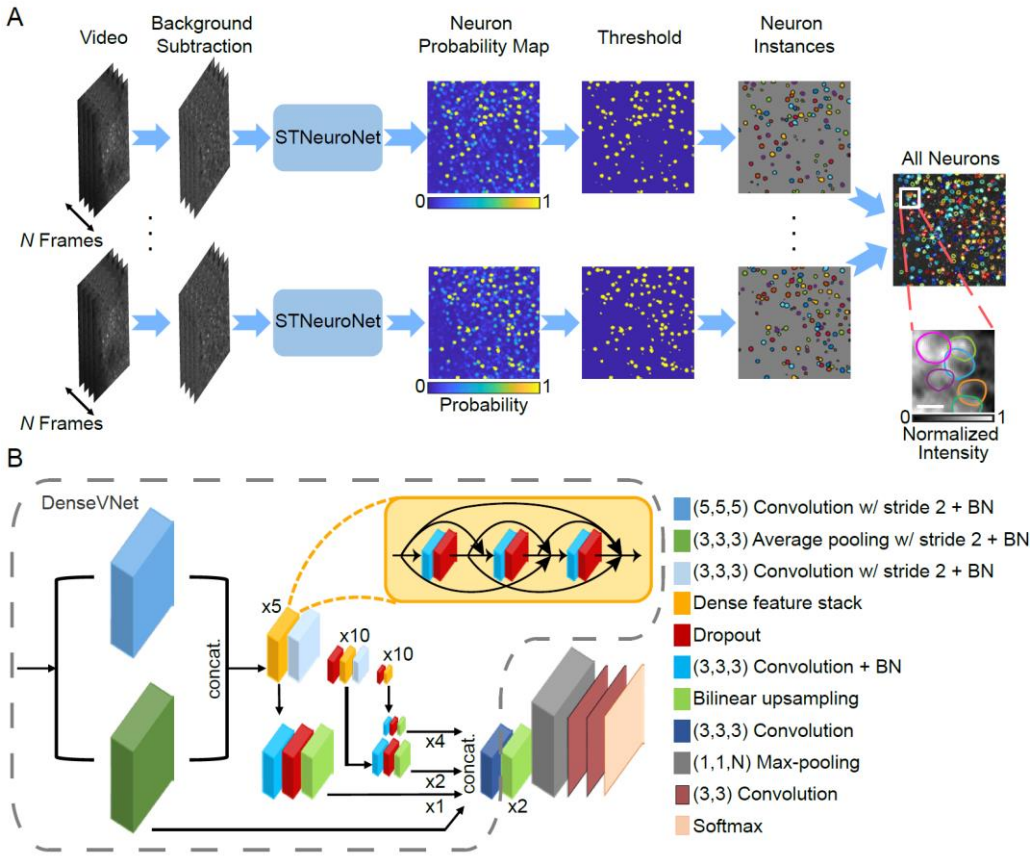


Fig. 2. Schematic for the proposed spatio-temporal deep-learning based segmentation of active neurons in two-photon calcium videos. (*A*) After removing background non-uniformity of each video batch (N = 120 frames), STNeuroNet predicts the neuron probability map. We identify neuron instances in the binarized probability maps from multiple temporal batches, which we then fuse into the final set of active neurons for the entire video. The right inset is the mean image of the region enclosed by the white box, normalized to its maximum fluorescence value. Scale bar is 10 µm. (*B*) STNeuroNet architecture details. The network generates feature maps at three different resolutions with a cascade of dense feature stacks and strided convolutional layers. These features maps are fused and further processed by the subsequent convolutional layers. A final bilinear upsampling layer transforms the feature maps to the original image resolution. A max-pooling layer summarizes the features along the time dimension. Finally, two 2D convolutional layers generate the segmentation logits. All convolutional layers use the rectified linear unit activation function. Numbers on top of the dense feature stacks indicate the number of convolutional layers involved, and numbers for the bilinear upsampling blocks indicate the upsampling factor. BN: Batch normalization.

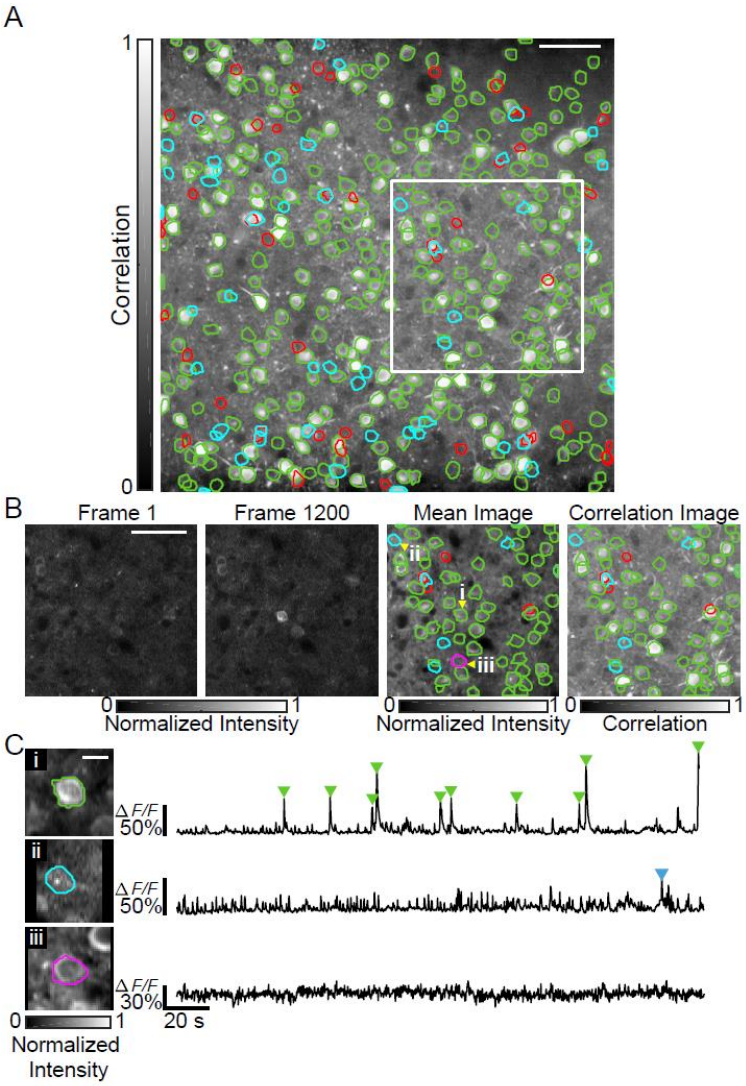


Fig. 3. STNeuroNet accurately identified active neurons from the Allen Brain Observatory dataset. (*A*) Detection results from 1200 frames (200 seconds) of a test video overlaid on the 200 × 200 pixels (156 μm × 156 μm) cropped region from the correlation image of the data. The neuron detection metrics (recall, precision, F_1) for the whole-size data are (0.86, 0.88, 0.87). *Green outlines*: true positives, *cyan outlines*: false negatives, and *red outlines*: false positives. Scale bar is 50 μm. (*B*) First and last frames, normalized mean image, and correlation image from the region enclosed in the white box in *A*. While many neurons are visible in the mean image, only active neurons were segmented (*green outlines*). The neuron marked with magenta is an example silent neuron that STNeuroNet effectively disregarded. Scale bar is 50 μm. (*C*) Example mean images of true positive, false negative, and silent neurons (*green*, *cyan*, and *magenta outlines*, respectively; *left*) and their time-series (*right*) from *B*. Scale bar is 10 μm.

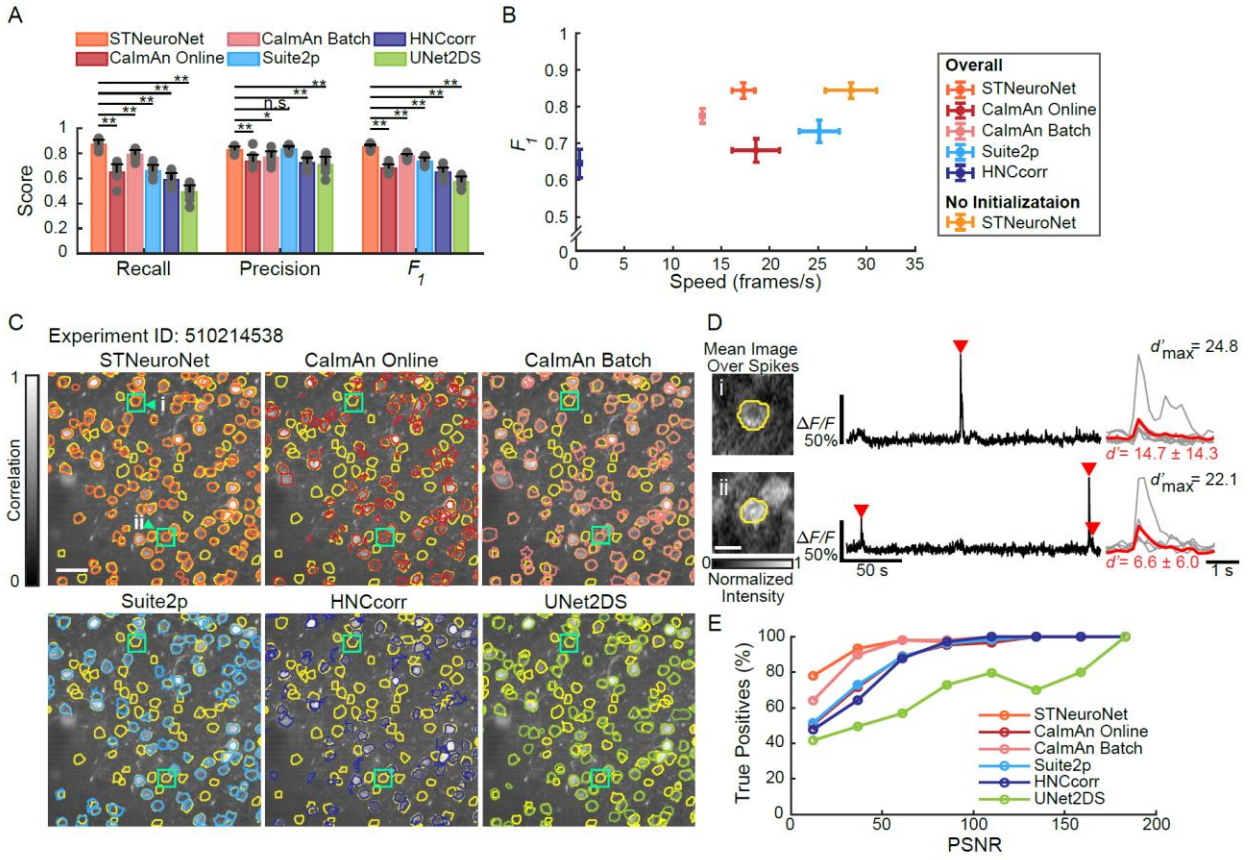


Fig. 4. STNeuroNet outperformed CaImAn, Suite2p, HNCcorr and UNet2DS on the Allen Brain Observatory dataset. (*A*) STNeuroNet's neuron identification score was superior to other methods (*: p-value<0.05 and **: p-value < 0.005, two-sided Wilcoxon rank sum test, n = 10 videos), which was largely due to its superior recall. (*B*) Our framework achieved superior detection performance over other methods at practically high processing speed. To facilitate visualization in this figure, we have excluded the relatively inaccurate yet fast UNet2DS (mean ± standard deviation of F_1 = 0.57 ± 0.04 and speed = 2263.3 ± 2.6 frames/s for n = 10 videos). Error bars in *A* and *B* are standard deviations for n = 10 videos. (*C*) Example data comparing the result of STNeuroNet to CaImAn (7), Suite2p (12), HNCcorr (15), and UNet2DS (18). The segmented neurons are marked with different colors for each algorithm on top of the correlation image, with the yellow markings denoting the GT neurons. Scale bar is 50 μm. (*D*) Example neurons from *C* identified by STNeuroNet and missed by other methods along with their time-series (*black traces*) and aligned activity-evoked signals (*gray traces*). Images on the left are the normalized mean images over the active intervals of the neurons. Traces are from a portion of the entire recording, with the times of putative calcium transients labeled with red markers. Red traces are the average of all aligned

transients. Scale bar is 10 μm. (*E*) Percentage of detected GT neurons versus binarized PSNR for all algorithms. The higher values of STNeuroNet's curve compared to other algorithms in the low PSNR regime indicate that our network identified a larger portion of neurons with low optical calcium response.

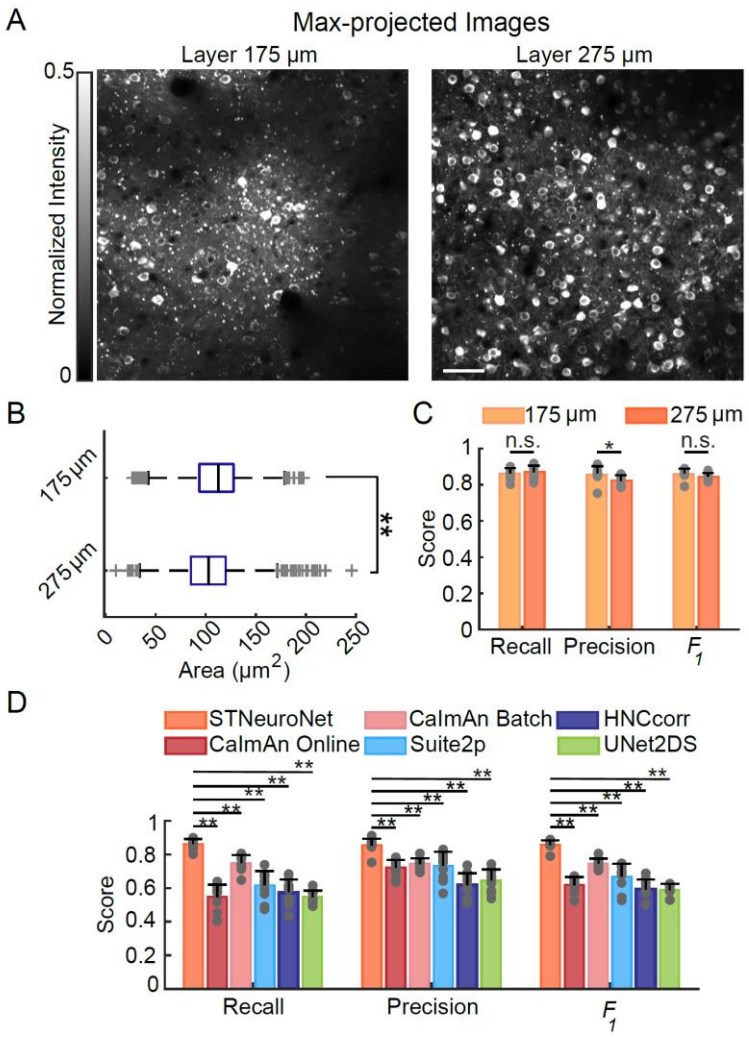


Fig. 5. Trained STNeuroNet performed equally well on data from a different cortical layer, outperforming CaImAn, Suite2p, HNCcorr and UNet2DS. (*A*) Qualitative comparison between Layer 275 µm and 175 µm data from the ABO dataset. Images are the normalized maximum-value projection images over the entire recording of two sample data. (*B*) The area of active neurons labeled from the two cortical depths were different (**: *p*-value < 0.005; *n* = 2182 and 3016 neurons from the 175 µm and 275 µm datasets, respectively), with the higher depth exhibiting smaller neurons. (*C*) The neuron detection scores were not significantly different for recall and F_1 (*p*-values = 0.5708 and 0.1212, respectively; *: *p*-value < 0.05; *n* = 10 videos for both groups) between the two datasets using the network trained on the 275 µm data to detect active neurons. (*D*) STNeuroNet's performance score on the 175 µm data was superior compared to other methods (*: *p*-value < 0.05 and **: *p*-value < 0.005; over *n* = 10 videos). All *p*-values were derived using the two-sided Wilcoxon rank sum test.

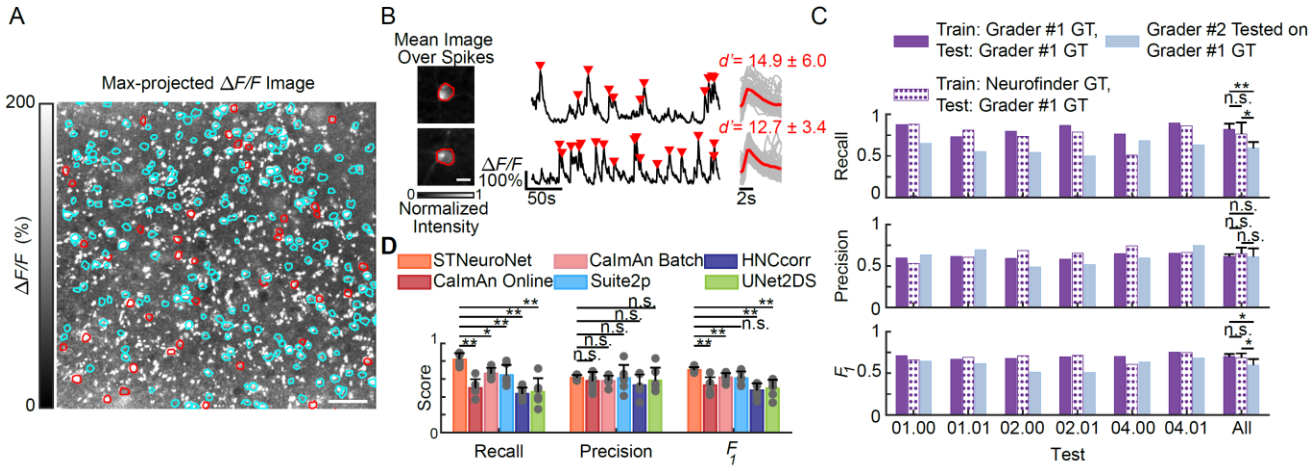


Fig. 6. STNeuroNet achieved best performance in the Neurofinder challenge, which contained suboptimal markings. (*A*) Overlay of the initial GT neurons (*cyan outlines*) and the added neurons after manual inspection (*red outlines*) on the maximum-projection of Δ*F*/*F* from the 04.00 training data. Scale bar is 50 µm. (*B*) Example neurons missed by the Neurofinder-supplied markings from *A* along with their neuropil-subtracted time-series (*black traces*). These neurons exhibit transients (*gray traces*: temporally aligned activity-evoked signals; *red traces*: average of gray traces) typical of GCaMP6-expressing neurons with high detection fidelity *d′* (reported values are mean ± standard deviation). The images on the left are the normalized average of frames within active time-intervals, defined as 0.5 seconds after the marked spike times. Scale bar is 10 µm. (*C*) When tested on grader #1's GT, STNeuroNet's performance was not significantly different when it was trained on either of Neurofinder's GT or grader #1's GT (*p*-value = 0.9372). Both networks achieved above-human performance in average F_1 score across the test dataset compared to grader #2, when tested with grader #1's GT (*p*-values = 0.0411 and 0.0087). (*D*) STNeuroNet statistically outperformed other methods except Suite2p on the Neurofinder test set, as denoted by the average F_1 score (*: *p*-value < 0.05 and **: *p*-value < 0.005). All *p*-values were derived using two-sided Wilcoxon rank sum test over $n = 6$ videos.

Supplementary Information for

Fast and robust active neuron segmentation in two-photon calcium imaging using spatio-temporal deep-learning

Somayyeh Soltanian-Zadeh, Kaan Sahingur, Sarah Blau, Yiyang Gong, and Sina Farsiu

Correspondence: Yiyang Gong and Sina Farsiu
Email: yiyang.gong@duke.edu and sina.farsiu@duke.edu

Supplementary Methods

Other algorithms used for comparison. *CaImAn.* We used the available code at https://github.com/flatironinstitute/CaImAn to implement the algorithm of (1). We selected the optimal parameter values for CaImAn Online and CaImAn Batch that resulted in the highest performance. Specifically, we performed a grid search over a range of values for the tuning parameters using leave-one-out cross-validation to quantify the performance on the ABO Layer 275 μm data. We reported the performance scores on the ABO Layer 175 μm test set and the Neurofinder test set using the best parameters determined by the ABO Layer 275 μm data and the Neurofinder training data, respectively. The CaImAn toolbox includes two pre-trained CNNs for the analysis of calcium imaging data. One CNN is used during the processing pipeline of the CaImAn Online method, and the other is used as a post-processing step to reduce falsely-detected masks. We have re-trained these two networks with the available data using the scripts provided by the authors.

When applying CaImAn Online to the ABO Layer 275 μm data, we changed the expected half-size of neurons from 5 pixels to 10 pixels (3.9 μm to 7.8 μm) and selected the number of components during the initialization phase from [2, 10, 50, 150] and the number of frames for initialization from [100, 200, 300]. We selected the minimum signal-to-noise ratio (SNR) for accepting new components from [2, 4, 6, 8], the maximum number of neurons added per frame from [5, 10, 25, 50], the threshold of the trained classifiers for adding new components during the online processing from [0.5, 0.7, 0.8, 0.9, 0.95], and the threshold for eliminating false positives

from 0 to 0.5 with step size of 0.01. When applying CaImAn Online to the Neurofinder dataset, we set the expected half-size of neurons, the initial batch size and the number of initialization components to the values used by (1). We selected the minimum acceptable SNR from [2, 2.5, 4, 6] and the maximum number of added neurons per frame from [5, 10, 20] while changing the classifier thresholds for adding new components and eliminating components from [0.5, 0.75, 0.8] and from 0 to 0.5 with step size of 0.1, respectively.

For CaImAn Batch on both ABO and Neurofinder datasets, we used the optimal half-size of neurons found from the CaImAn Online results. We set the patches to be 100×100 pixels with 10 pixels overlap between patches. We set the number of components per patch to 40, twice the maximum average number of neurons per 100×100 pixels area from the GT set, to avoid low recall. We selected the spatial correlation threshold from [0.75, 0.80, 0.85], the upper and lower thresholds for the CNN classifier from [0.8, 0.9, 0.95, 0.98] and 0 to 0.5 with step size of 0.1, respectively. We selected the minimum SNR for the ABO dataset from 4 to 10 with increment of 2, and for the Neurofinder dataset from [1.8, 2, 2.5, 3]. We used the optimal values that yielded the highest mean F_1 score across the training set to quantify the final performances. As in (1), we binarized each real-valued detected mask by using 0.2 times the maximum value of the mask as the threshold.

Suite2p. We used the code provided by (2) available online at https://github.com/cortex-lab/Suite2P. Through leave-one-out cross-validation, we quantified the performance of Suite2p on the ABO Layer 275 µm dataset. We used all of the ABO Layer 275 µm data and Neurofinder training data to quantify the performance on the ABO Layer 175 µm test set and the Neurofinder test set, respectively. For both the ABO and Neurofinder datasets, we varied the diameter of neurons from 7.8 µm to 15.6 µm with step size of 3.9 µm, the number of singular value decomposition (SVD) components from 200 to 800 with step size of 100, number of frames for SVD from 1000 to 4000 in steps of 1000. We selected the probability threshold of their ROI classifier from 0 to 0.5 with step size of 0.1, and the minimum and maximum acceptable sizes from 15 to 120 μm^2 and 100 to 845 μm^2 in increments of 18 μm^2 and 426 μm^2, respectively. We kept all other parameters at the default values set by the authors of (2). For each data, we ran the Suite2p procedure until the number of detected neurons did not change, or until we reached a maximum of one hundred iterations. We also trained their ROI classifier on the training videos by

manually curating the results that yielded the largest number of detected neurons. For each validation iteration, we used the best combination of parameters that yielded the highest mean F_1 score on the training data for the test data to report the final performance scores of Suite2p.

HNCcorr. HNCcorr is a graph-cut based method that processes the correlation image. We used the code provided by (3) at https://github.com/hochbaumGroup/HNCcorr. Like other methods, we performed leave-one-out cross-validation to quantify HNCcorr's performance on the ABO Layer 275 µm data, and used all of the ABO Layer 275 µm data and the Neurofinder training data to quantify the performance on the ABO Layer 175 µm test set and the Neurofinder test set, respectively. For the ABO dataset, we set the segmentation window size to 37 pixels (28.9 µm) and the average neuron size to 107.5 μm^2 We selected the percentage of seeds from 0.1 to 0.7 with step size of 0.1, the seed size from 1×1 pixel to 5×5 pixels, and the minimum and maximum acceptable sizes from 35 to 60 μm^2 and 122 to 243 μm^2 with step size of 6 μm^2 and 30 μm^2, respectively. For the Neurofinder dataset, based on the parameters reported in (3), we set the segmentation window size to 41 pixels, the percentage of seeds to 0.4, and the average neuron sizes to the values reported in Supplementary Table 3 of (3). In accordance with the values used in (3), we changed the seed size from 1×1 pixel to 5×5 pixels, and the minimum and maximum acceptable sizes from 30 to 50 pixels and 200 to 800 pixels with step size of 10 and 100 pixels, respectively. We used the combination of parameters that yielded the highest mean F_1 score on the training data for the test data to report the final performance scores of HNCcorr.

UNet2DS. We used the code provided by (4) at https://github.com/alexklibisz/deep-calcium to train and test the UNet2DS network on the ABO dataset. This CNN is based on the popular UNet (5) and uses the mean image of the data to segment neurons. We performed leave-one-out cross-validation to quantify the performance of this network on the ABO Layer 275 µm data, and used the ABO Layer 275 µm data and the Neurofinder training data to quantify the performance on the ABO Layer 175 µm test set and Neurofinder test set, respectively. Using the same training procedure outlined by (4), we trained UNet2DS for 50 epochs with 100 training iterations in each epoch using sixteen randomly cropped 128×128 pixels regions from the mean image, utilizing the dice-loss and the Adam optimizer. In accordance with (4), we tracked the F_1 score on a validation video selected from the training set to ensure the network was not overfitting. For the Neurofinder data, we used the exact scripts provided by (4) to train on the six training videos. At inference

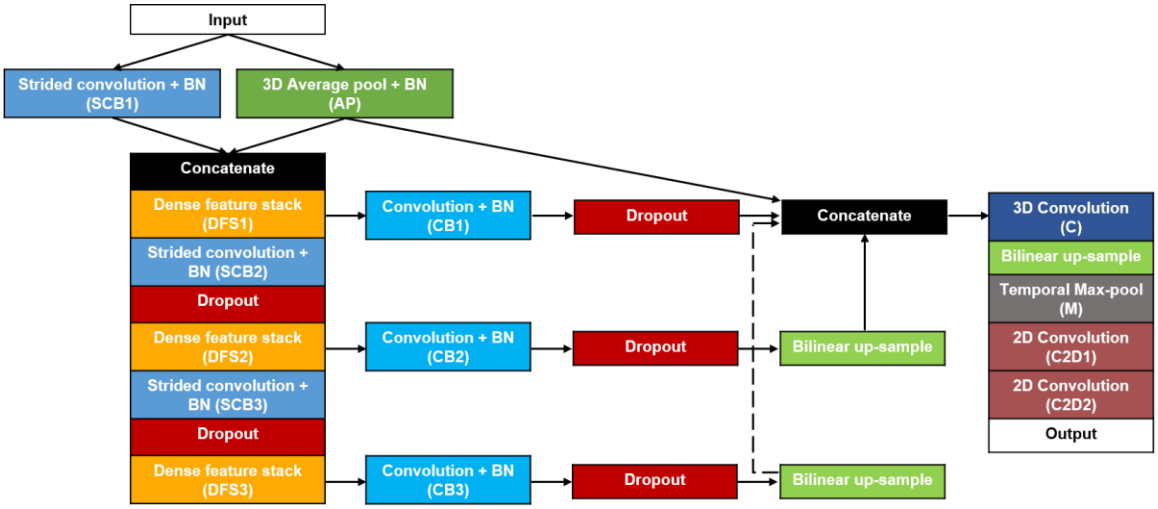
time, we averaged the predictions from eight rotations and reflections of the full spatial-extent of the test image to make the final prediction. We used the combination of parameters that yielded the highest mean F_1 score on the training data for the test data to report the final performance scores.

Supplementary Experiment

Out of an abundance of caution, we set out a sanity check experiment to show that our algorithm's superior performance was not by luck or design tailored to the specific GT created by graders #1 and #2 (what we referred to as the final GT previously in the main text). To do this, we used grader #3's marking as the final GT set for training and testing on the Layer 275 μm dataset through leave-one-out cross-validation. Recall that grader #3 marked the data from scratch (i.e. was blinded to the initial masks from the Allen Institute). As the results show in the following table, our framework outperformed all other algorithms in terms of the F_1 score (p-values < 0.02, two-sided Wilcoxon rank sum test over n = 10 videos) and was on par with human performance (compared to the GT of graders #1 and #2; p-value = 0.970, two-sided Wilcoxon rank sum test over n = 10 videos). Compared to Table S2, the performance of all algorithms decreased. This reduced performance is expected because the GT used in Table S2 was based on the markings of two graders as judged by the senior and more experienced grader (grader #1) and thus is expected to be more consistent than the new GT which was only from one grader (grader #3).

STNeuroNet outperformed other methods when using grader #3 as the final GT. Reported numbers are in F_1 (Recall, Precision) format, where in each field we report the mean ± standard deviation across n = 10 videos.

STNeuroNet	*CaImAn Online*	*CaImAn Batch*	*Suite2p*	*HNCcorr*	*UNet2DS*	*GT from Graders #1 and #2*
0.78±0.03 (0.79±0.07, 0.77±0.04)	0.65±0.06 (0.62±0.07, 0.70±0.05)	0.72±0.05 (0.75±0.07, 0.69±0.05)	0.71±0.06 (0.62±0.08, 0.83±0.04)	0.64±0.05 (0.58±0.09, 0.72±0.02)	0.56±0.01 (0.49±0.04, 0.66±0.06)	0.78±0.03 (0.82±0.06, 0.75±0.02)



Layer	*Input*	*Output*	*Kernel*	*Stride*	*Subunits* $n \times n_c$
AP	$144 \times 144 \times 120 \times 1$	$72 \times 72 \times 60 \times 1$	$3 \times 3 \times 3$	2	
SCB1	$144 \times 144 \times 120 \times 1$	$72 \times 72 \times 60 \times 24$	$5 \times 5 \times 5$	2	
DFS1	$72 \times 72 \times 60 \times 25$	$72 \times 72 \times 60 \times 20$	$3 \times 3 \times 3$	1	5×4
SCB2	$72 \times 72 \times 60 \times 20$	$36 \times 36 \times 30 \times 24$	$3 \times 3 \times 3$	2	
DFS2	$36 \times 36 \times 30 \times 24$	$36 \times 36 \times 30 \times 80$	$3 \times 3 \times 3$	1	10×8
SCB3	$36 \times 36 \times 30 \times 80$	$18 \times 18 \times 15 \times 24$	$3 \times 3 \times 3$	2	
DFS3	$18 \times 18 \times 15 \times 24$	$18 \times 18 \times 15 \times 160$	$3 \times 3 \times 3$	1	10×16
CB1	$72 \times 72 \times 60 \times 20$	$72 \times 72 \times 60 \times 12$	$3 \times 3 \times 3$	1	
CB2	$36 \times 36 \times 30 \times 80$	$36 \times 36 \times 30 \times 24$	$3 \times 3 \times 3$	1	
CB3	$18 \times 18 \times 15 \times 160$	$18 \times 18 \times 15 \times 24$	$3 \times 3 \times 3$	1	
C	$72 \times 72 \times 60 \times 61$	$72 \times 72 \times 60 \times 10$	$3 \times 3 \times 3$	1	
M	$144 \times 144 \times 120 \times 10$	$144 \times 144 \times 1 \times 10$	$1 \times 1 \times 120$	1	
C2D1	$144 \times 144 \times 1 \times 10$	$144 \times 144 \times 1 \times 10$	3×3	1	
C2D2	$144 \times 144 \times 1 \times 10$	$144 \times 144 \times 1 \times 2$	3×3	1	

Fig. S1. Detailed parameters for STNeuroNet architecture. Dimensions are written as: $S_x \times S_y \times N \times C$, where S_x and S_y are the x and y spatial sizes, N is the number of frames, and C is the number of channels. Subunits reports the number of convolutional layers in the dense feature stacks (n) and the number of channels in each of these layers (n_c).

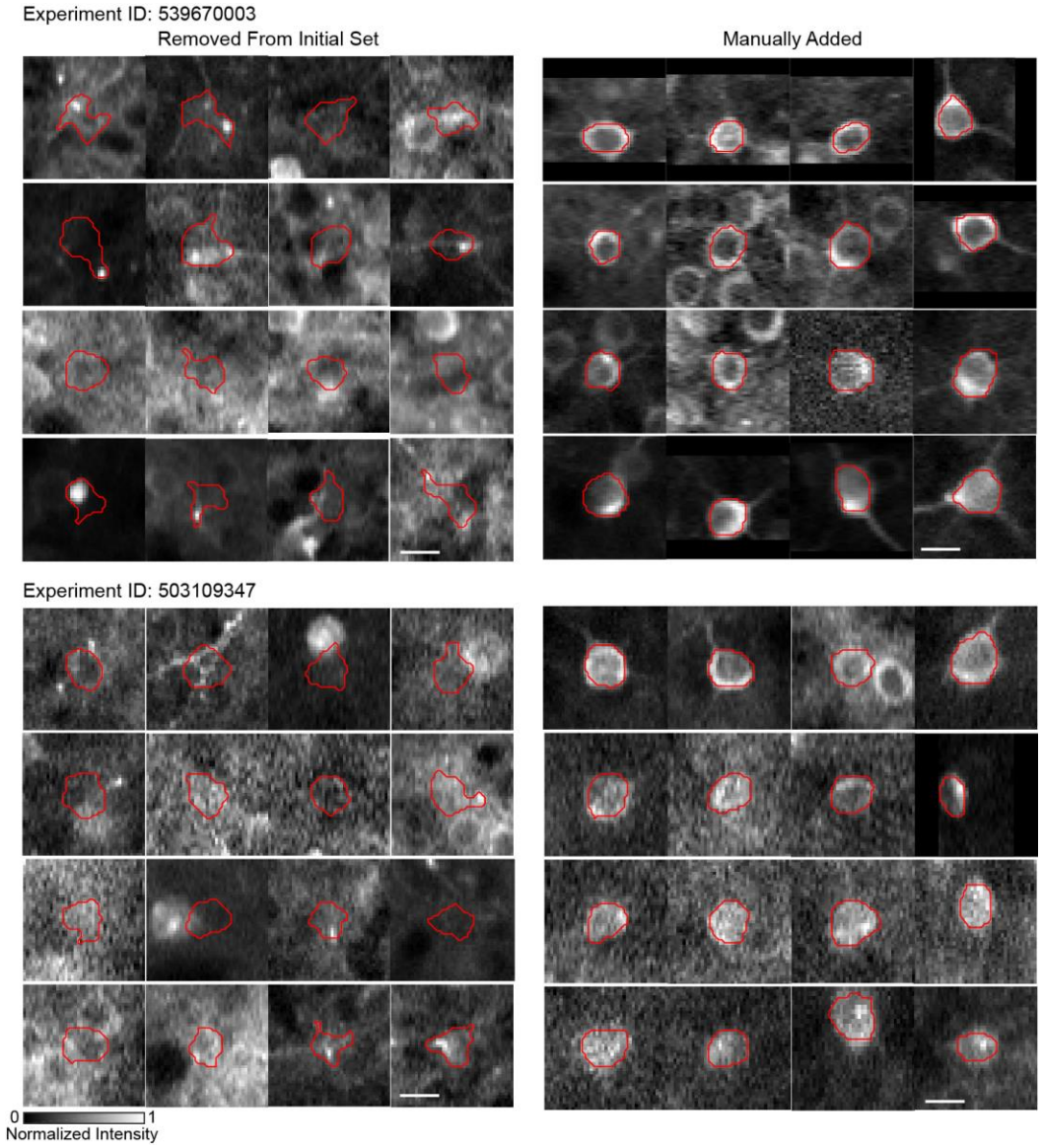


Fig. S2. Representative examples of active neuron labeling errors in the Allen Brain Observatory dataset. Example cases that were (*left*) removed from the initial set of marking accompanied with the Allen Brain Observatory dataset and (*right*) manually added by graders to produce the final ground truth set. Images are normalized mean of video frames at peak signal time-points. Scale bars are 10 µm.

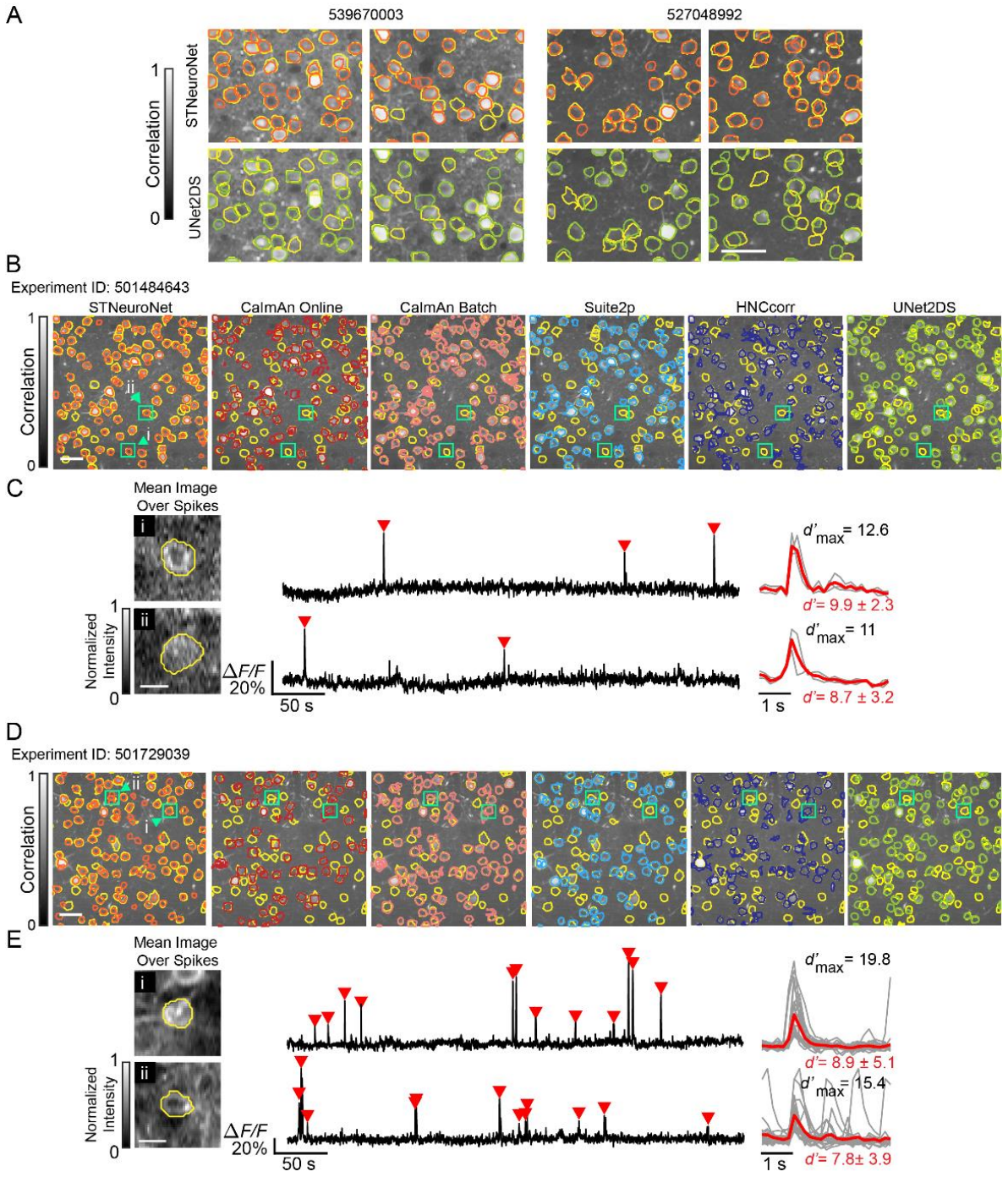


Fig. S3. STNeuroNet outperforms CaImAn, Suite2p, HNCcorr and UNet2DS on the Allen Brain Observatory dataset. Related to Fig. 4. (*A*) UNet2DS cannot separate overlapping neurons, resulting in low recall scores. Scale bar is 50 µm. (*B-E*) Example neurons from different data identified by STNeuroNet and missed by all other methods. (*B*) and (*D*) highlight individual neurons (i and ii) within the population markings, while (*C*) and (*E*) plot the time-series of the

highlighted neurons (*black traces*) and aligned activity-evoked signals (*gray traces*). The segmented neurons are marked with different colors for each algorithm with yellow markings denoting GT neurons. All images are 300 × 300 pixels (234 μm × 234 μm) images cropped from the center of the data. (*C, E*) Images on the left are the normalized mean images over the spike intervals of the neurons. Traces are from a portion of the entire recording, with the spike times of the neuron labeled with red markers. Red traces are the average of each set of corresponding gray traces. Scale bars are 50 and 10 μm for the large population-scale images and small single-neuron images, respectively.

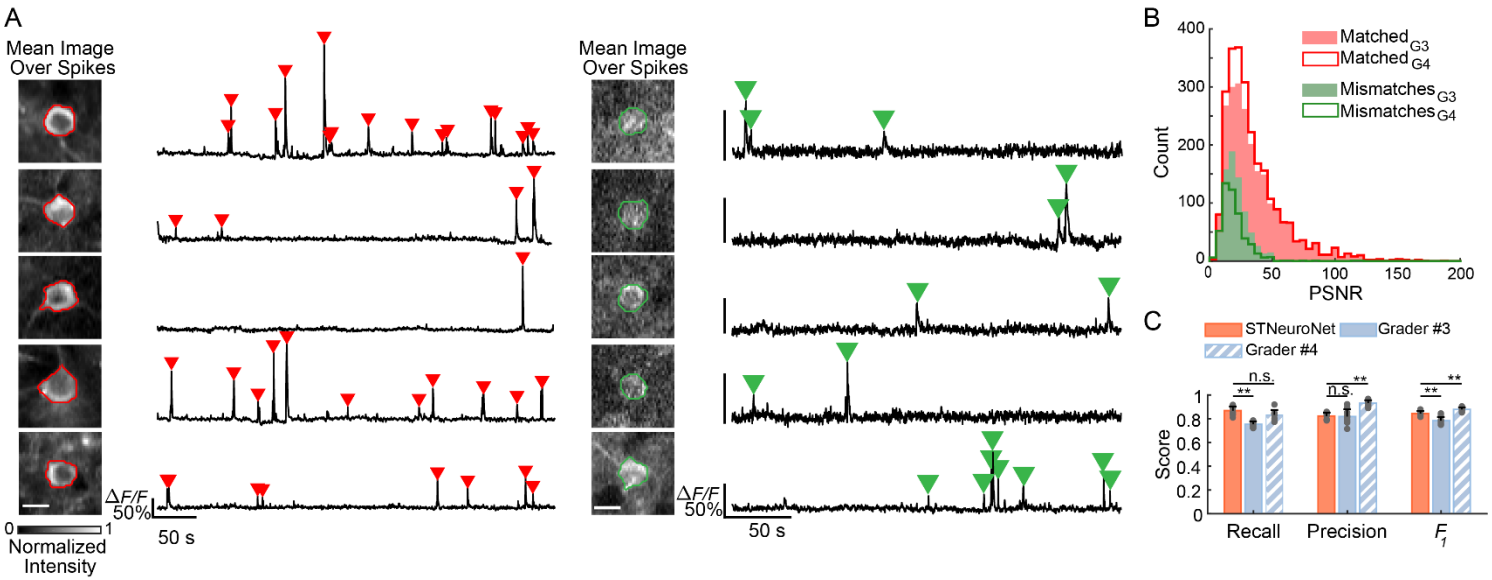


Fig. S4. Inter-human agreement test for Allen Brain Observatory neuron segmentation. Related to Fig. 5. (*A*) Examples of common neurons between GT and grader # 3 (*red* data) and missed neurons by grader #3 (*green* data). Images are the normalized mean image of the neurons over their active time-intervals, defined as 0.5 seconds after the marked spike times. Missed neurons exhibit low peak $\Delta F/F$ or atypical appearance. Scale bars are 10 µm. Time-series correspond to a portion of the data in time. (*B*) Histogram of the PSNR for mismatched (*green* data) and matched neurons (*red* data) between GT and graders #3 and #4. (*C*) Our algorithm achieved similar recall score as grader #4 (p-value = 0.0757) and similar precision as grader #3 (p-value = 0.6776) resulting in a F_1 score between that of grader #3 and grader #4. All p-values were calculated using the two-sided Wilcoxon rank sum test for n = 10 videos (n.s.: not significant; **: p-value < 0.005).

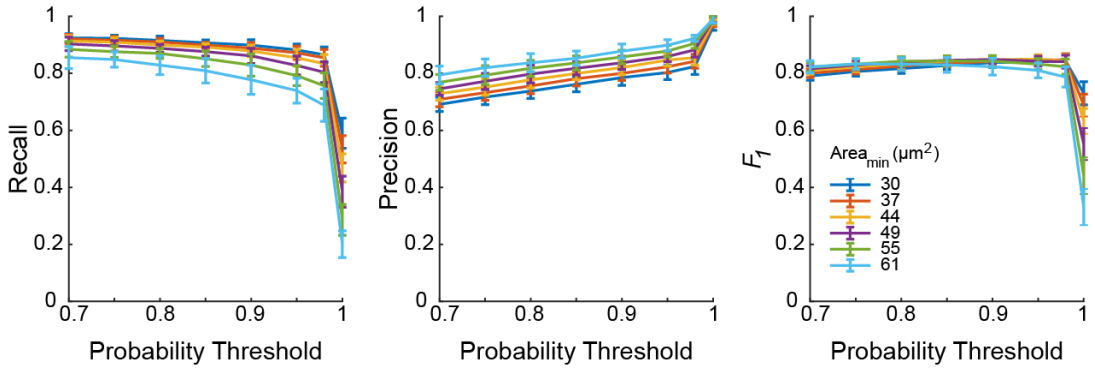


Fig. S5. The optimal thresholds in the post-processing step of our algorithm are determined through leave-one-out cross-validation. Example results of recall, precision, and F_1 cores by applying different levels of probability and minimum area thresholds to $n = 9$ training videos from the Allen Brain Observatory dataset. For this example, the optimal thresholds for the probability map and minimum area were 0.95 and 44 μm^2, respectively.

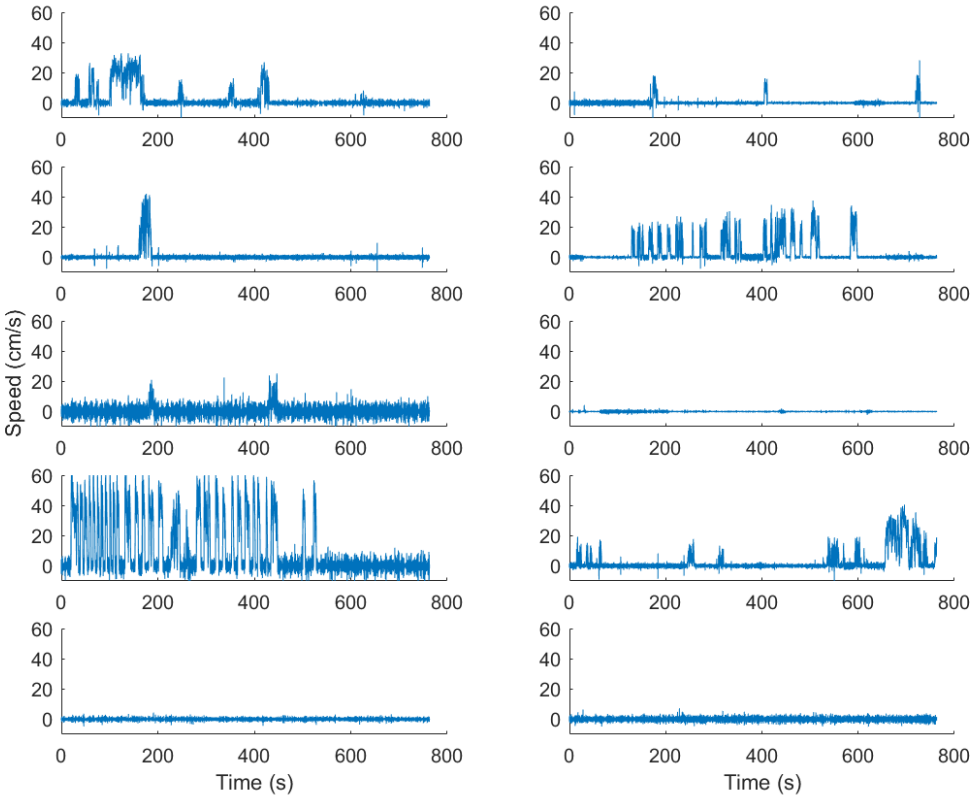


Fig. S6. Majority of selected mice for data analysis had low running speed during the data time-interval. Nine of the mice in the selected Allen Brain Observatory dataset were stationary (speed < 1 cm/s) for more than 50% of the time-interval while a drifting grating stimulus was presented (Allen Institute of Brain Science, June 2017).

Table S1. Number of neurons found by each grader, the final ground truth (GT) set, the original labels, and all methods for the test datasets. The Final GT was determined by graders #1 and #2 for the Allen Brain Observatory dataset, and by grader #1 for the Neurofinder dataset. The number of neurons found by all algorithms for the ABO Layer 275 μm data are from the leave-one-out cross-validation tests, while the number of neurons for the ABO Layer 175 μm data are from the algorithms trained/tuned on the ABO Layer 275 μm data. The reported number of neurons for the Neurofinder test set are from training/tuning each algorithm on the Neurofinder training set**.**

Data ID	*Grader1*	*Grader2*	*Grader3*	*Grader4*	*Final GT*	*Original*	*STNeuroNet*	*CaImAn Online*	*CaImAn Batch*	*Suite2p*	*HNCcorr*	*UNet2DS*
Layer 275 μm												
534691284	372	244	310	292	355	264	382	381	389	303	316	163
531006860	294	208	243	252	282	230	303	240	299	241	217	197
502608215	338	267	272	290	322	318	338	292	316	263	301	219
503109347	351	282	319	295	331	353	338	293	361	242	260	217
501484643	269	192	250	215	269	184	261	256	284	201	267	198
501574836	300	217	278	245	278	240	293	241	235	222	244	195
501729039	248	212	216	220	235	227	266	134	220	160	212	212
539670003	401	224	331	399	386	378	423	356	401	305	351	237
510214538	313	302	278	276	306	284	293	277	322	236	250	227
527048992	263	242	277	227	252	211	289	241	309	230	212	202
Layer 175 μm												
501704220	163	177	-	-	176	142	175	140	199	226	135	176
501271265	242	216	-	-	237	215	243	180	237	245	234	194
501836392	187	190	-	-	198	178	209	156	199	211	201	194
502115959	210	214	-	-	227	205	232	181	233	242	215	186
502205092	264	268	-	-	273	292	278	225	288	269	283	205
504637623	198	207	-	-	205	197	200	206	202	219	252	177
510514474	200	226	-	-	209	226	214	152	201	208	184	180
510517131	230	241	-	-	249	266	252	204	264	226	201	195
540684467	249	215	-	-	247	290	217	143	207	282	243	171
545446482	157	158	-	-	161	171	178	87	163	177	117	156
Neurofinder Test												
01.00.test	184	189	-	-	184	-	270	125	189	255	193	256
01.01.test	123	98	-	-	123	-	146	119	145	251	84	105
02.00.test	206	229	-	-	206	-	277	163	216	183	143	147
02.01.test	160	155	-	-	160	-	239	180	215	159	140	146
04.00.test	149	171	-	-	149	-	176	112	188	178	160	84
04.01.test	330	278	-	-	330	-	453	320	362	247	234	148

Table S2. Summary of performances on all datasets. Reported numbers are in F_1 (Recall, Precision) format, where in each field we report the mean ± standard deviation across n = 10 and n = 6 videos for the ABO and Neurofinder dataset, respectively.

Method / *Test*	*STNeuroNet*	*CaImAn Online*	*CaImAn Batch*	*Suite2p*	*HNCcorr*	*UNet2DS*
ABO Layer 275 (cross-validation)	0.84±0.02 (0.87±0.04, 0.82±0.03)	0.68±0.03 (0.64±0.07, 0.73±0.06)	0.77±0.02 (0.79±0.04, 0.76±0.05)	0.73±0.03 (0.66±0.05, 0.83±0.03)	0.65±0.04 (0.59±0.05, 0.72±0.04)	0.57±0.04 (0.49±0.06, 0.71±0.07)
ABO Layer 175[a]	0.86±0.03 (0.86±0.03, 0.85±0.04)	0.62±0.05 (0.55±0.07, 0.72±0.05)	0.75±0.03 (0.75±0.05, 0.74±0.03)	0.67±0.08 (0.62±0.09, 0.73±0.08)	0.59±0.06 (0.58±0.08, 0.62±0.07)	0.59±0.04 (0.55±0.04, 0.65±0.07)
Neurofinder Test[b]	0.70±0.03 (0.82±0.07, 0.61±0.03)	0.53±0.09 (0.50±0.10, 0.58±0.10)	0.62±0.05 (0.67±0.06, 0.58±0.06)	0.61±0.08 (0.64±0.12, 0.61±0.15)	0.47±0.08 (0.43±0.07, 0.53±0.12)	0.49±0.10 (0.46±0.15, 0.58±0.14)

[a]: ABO Layer 275 used for training

[b]: Trained with Grader 1 labels with single optimization over all Neurofinder Train data. Evaluated with Grader 1 Test label

Table S3. Inter-human agreement test reflects a bias within graders #1, #2, and #4 due to the available pilot segmentation labels. Performance of all graders and STNeuroNet on Allen Brain Observatory Layer 275 μm data with markings from different graders serving as the evaluation ground-truth (GT). The final GT for training STNeuroNet was the set of consensus markings of graders #1 and #2. The reported numbers are in F_1 (Recall, Precision) format, where in each field we report the mean ± standard deviation across n = 10 videos**.**

Test / *GT*	*Grader #1*	*Grader #2*	*Grader #3*	*Grader #4*	*STNeuroNet*
Grader #1	N/A	0.81±0.05 (0.72±0.09, 0.94±0.04)	0.78±0.03 (0.73±0.02, 0.83±0.06)	0.88±0.02 (0.82±0.04, 0.96±0.02)	0.84±0.02 (0.84±0.04, 0.83±0.03)
Grader #2	0.81±0.05 (0.94±0.04, 0.72±0.09)	N/A	0.73±0.04 (0.79±0.06, 0.68±0.07)	0.84±0.05 (0.90±0.05, 0.80±0.10)	0.74±0.04 (0.86±0.05, 0.65±0.08)
Grader #3	0.78±0.03 (0.83±0.06, 0.73±0.02)	0.73±0.04 (0.68±0.07, 0.79±0.06)	N/A	0.78±0.03 (0.77±0.07, 0.79±0.04)	0.75±0.03 (0.81±0.06, 0.71±0.03)
Grader #4	0.88±0.02 (0.96±0.02, 0.82±0.04)	0.84±0.05 (0.80±0.10, 0.90±0.05)	0.78±0.03 (0.79±0.04, 0.77±0.07)	N/A	0.80±0.03 (0.87±0.03, 0.74±0.05)

Table S4. Description of data used from the Allen Brain Observatory. All data are from the primary visual cortex**.**

Cortical Layer	Transgenic Line	Experiment ID	Cortical Layer	Transgenic Line	Experiment ID
275 µm	Cux2-CreERT2-Cre	539670003	175 µm	Cux2-CreERT2-Cre	501704220
		531006860			501836392
		501574836			510514474
		501484643			504637623
		503109347			501271265
		534691284			502115959
		502608215			502205092
		501729039			510517131
	Rorb-IRES2-Cre	510214538		Emx1-IRES-Cre	540684467
		527048992			545446482

Table S5. STNeuroNet performance on all data when trained on different datasets. Reported numbers are in F_1 (Recall, Precision) format, where in each field we report the mean ± standard deviation across n = 10 and n = 6 videos for the ABO and Neurofinder datasets, respectively**.**

Train / *Test*	ABO Layer 275 μm	Neurofinder Train	ABO Layer 275 μm and Neurofinder Train	All ABO (Layer 275 μm and 175 μm)
ABO Layer 275 μm	0.84±0.02 [a] (0.87±0.04, 0.82±0.03)	0.74±0.04 (0.86±0.02, 0.64±0.05)	N/A	N/A
ABO Layer 175 μm	0.86±0.03 (0.86±0.03, 0.85±0.04)	0.74±0.04 (0.82±0.05, 0.68±0.05)	0.85±0.03 (0.88±0.03, 0.82±0.05)	N/A
Neurofinder Test	0.62±0.17 (0.52±0.24, 0.88±0.08)	0.70±0.03 (0.82±0.07, 0.61±0.03)	0.75±0.04 (0.72±0.11, 0.79±0.04)	0.67±0.11 (0.60±0.21, 0.83±0.09)
Neurofinder Train	0.48±0.13 (0.37±0.18, 0.83±0.13)	N/A	N/A	0.55±0.11 (0.45±0.18, 0.80±0.13)

[a]: Performance quantified with leave-one-out cross-validation.

Supplementary References

1. Giovannucci A*, et al.* (2019) CaImAn: An open source tool for scalable Calcium Imaging data Analysis. *eLife* 8:e38173.
2. Pachitariu M*, et al.* (2017) Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *BioRxiv*:061507.
3. Spaen Q, Hochbaum DS, & Asín-Achá R (2017) HNCcorr: A Novel Combinatorial Approach for Cell Identification in Calcium-Imaging Movies. *arXiv preprint arXiv:1703.01999*.
4. Klibisz A, Rose D, Eicholtz M, Blundon J, & Zakharenko S (2017) Fast, Simple Calcium Imaging Segmentation with Fully Convolutional Networks. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, (Springer), pp 285-293.
5. Ronneberger O, Fischer P, & Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer), pp 234-241.