



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

Somayyeh Soltanian-Zadeh^a, Kaan Sahingur^a, Sarah Blau^a, Yiyang Gong^{a,b,1}, and Sina Farsiu^{a,c,1}

^aDepartment of Biomedical Engineering, Duke University, Durham, NC 27708; ^bDepartment of Neurobiology, Duke University, Durham, NC 27708; and ^cDepartment of Ophthalmology, Duke University Medical Center, Durham, NC 27710

Edited by Terrence J. Sejnowski, Salk Institute for Biological Studies, La Jolla, CA, and approved March 19, 2019 (received for review July 28, 2018)

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 spatiotemporal information in two-photon calcium imaging movies, we propose a 3D 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.

deep learning | calcium imaging | neuron segmentation | two-photon microscopy | open source

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- to 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 carries 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 nonnegative 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 CNNs 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 with 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 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

Significance

Two-photon calcium imaging is a standard technique of neuroscience laboratories 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 toward 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.

Author contributions: S.S.-Z., Y.G., and S.F. designed research; S.S.-Z., Y.G., and S.F. performed research; S.S.-Z., K.S., S.B., Y.G., and S.F. analyzed data; and S.S.-Z., Y.G., and S.F. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Published under the PNAS license.

Data deposition: 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>).

¹To whom correspondence may be addressed. Email: yiyang.gong@duke.edu or sina.farsiu@duke.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1812995116/-DCSupplemental.

Published online April 11, 2019.

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 nonactive 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 with the background after undergoing averaging to the mean image. Finally, 2D segmentation of mean images has difficulty in delineating the neuron boundaries between overlapping neurons that independently fire in time (Fig. 1).

Three-dimensional 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 with methods that process 2D images, spatiotemporal methods can provide more accurate results in identifying sparsely spiking and overlapping neurons but are also computationally more challenging (13). Compared with iterative methods such as CNMF, a 3D CNN architecture could produce high computational efficiency for long-duration, large-scale recordings. Three-dimensional 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 (GT) markings. Indeed, the Allen Brain Observatory (ABO) (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 GT, 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 CNN-based method with spatiotemporal 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 (28).

Results

Spatiotemporal Neuron Segmentation Using Deep Learning. The key feature of our active neuron segmentation framework (Fig. 2A) was a 3D CNN architecture that we named Spatiotemporal NeuroNet (STNeuroNet) (Fig. 2B and *SI Appendix*, Fig. S1). The 3D con-

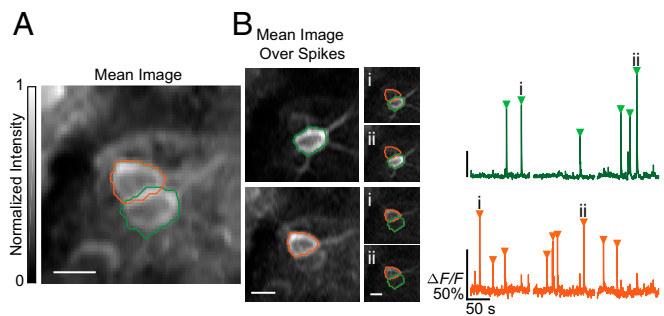


Fig. 1. Overlapping neurons complicate active neuron segmentation. (A) Neurons can have overlapping regions in two-photon calcium imaging data due to the projection of a 3D volume onto a 2D imaging plane, as evident in the mean image, normalized to the maximum intensity of the cropped region. (B) The 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 s after the marked spike times). We separated traces of these overlapping neurons using the linear regression approach of the Allen Institute (30). (Scale bars, 10 μm .)

volitional layers in STNeuroNet extracted local spatiotemporal 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 spatiotemporal 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. Finally, 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 ABO Dataset.

We first quantified the performance of our method using a subset of the ABO dataset. This dataset included the first 12 min and 51 s of two-photon microscopy recordings from 275 μm deep in the primary visual cortex (VISp) of 10 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, 29) 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 (30). We inspected the provided set of masks and found that some of the masks did not correspond to active neurons in the selected ~ 13 -min 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 GT labels (*Methods* and *SI Appendix*, Table S1). Overall, we removed $n = 40 \pm 23.6$ masks (mean \pm SD over $n = 10$ videos) from the initial ABO marking as they were not located on the soma of active neurons, accounting for $13.9 \pm 5.7\%$ of the initial neurons, and added $n = 72.7 \pm 20.9$ neurons, accounting for $24.2 \pm 5.9\%$ of the final GT neurons. The final set of neurons comprising the GT demonstrated peak calcium responses with $d' \geq 4$ within the spike

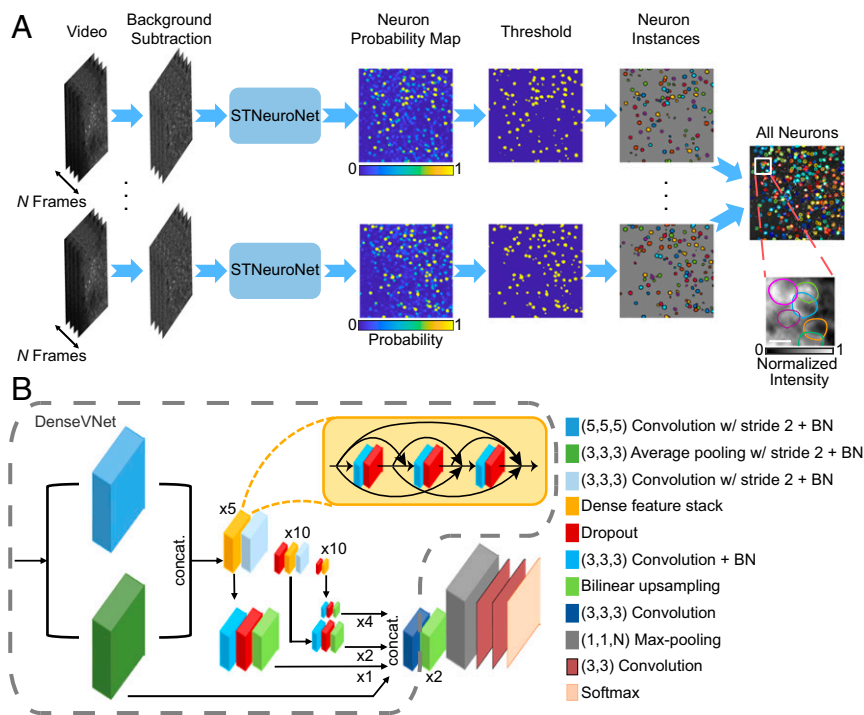


Fig. 2. Schematic for the proposed spatiotemporal deep learning based segmentation of active neurons in two-photon calcium videos. (A) After removing background nonuniformity of each video batch ($n = 120$ frames), STNeuroNet predicts the neuron probability map. We identify neuron instances in the binarized probability maps from multiple temporal batches, which we then fuse into the final set of active neurons for the entire video. (Right Inset) The mean image of the region enclosed by the white box, normalized to its maximum fluorescence value. (Scale bar, $10 \mu\text{m}$.) (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.

detection formalism (*Methods*), which were at significantly higher levels compared with the distribution of d' values from the baseline due to noise ($P < 0.001$, one-sided Z-test using $n = 500$ baseline samples for each of the 3,016 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 $144 \times 144 \times 120$ segments of input data took 11.5 h for 36,000 iterations. After training, STNeuroNet generated neuron predictions in 171.24 ± 21.28 s (mean \pm SD over $n = 10$ videos) when processing $4,624 \pm 5$ 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 per s (mean \pm SD 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 five times better than the reported number.

Fig. 3 shows an illustrative example of our framework applied on a time interval of $n = 1,200$ 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. 3B. To better illustrate temporal neuronal activity, we also show the correlation image, defined as the mean correlation value between each pixel with its four-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. 3B and C).

Using the same 10 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 ($P < 0.005$, two-sided Wilcoxon rank sum test over $n = 10$ videos; Fig. 4A and *SI Appendix, Table S2*) at higher speed compared with CaImAn Batch and HNCcorr ($P < 0.005$, two-sided Wilcoxon rank sum test over $n = 10$ videos), while being as fast as CaImAn Online and slower than Suite2p ($P = 0.3075$ and < 0.005 , respectively; two-sided Wilcoxon rank sum test over $n = 10$ videos; Fig. 4B) when processing 487×487 -pixel videos. After disregarding the initialization time of STNeuroNet, our framework was significantly faster than Suite2p ($P = 0.026$, 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 = $2,263.3 \pm 2.6$ frames per s for $n = 10$ videos), but it was not able to separate overlapping neurons, resulting in low recall values compared with other methods (*SI Appendix, Fig. S3A*).

We further investigated the underlying source for our framework's superior recall compared with other spatiotemporal methods. Fig. 4C and D and *SI Appendix, Fig. S3B-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. 4E. STNeuroNet's higher percentage of true positive neurons compared with 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 $22.4 \pm 7.5\%$, $7.9 \pm 3.6\%$, $21.0 \pm 4.8\%$, $26.1 \pm 4.6\%$, and $38.1 \pm 5.9\%$ more neurons (mean \pm SD for $n = 10$ videos) from the GT compared with 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 interhuman agreement test. Grader 3 labeled 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. S4A* and

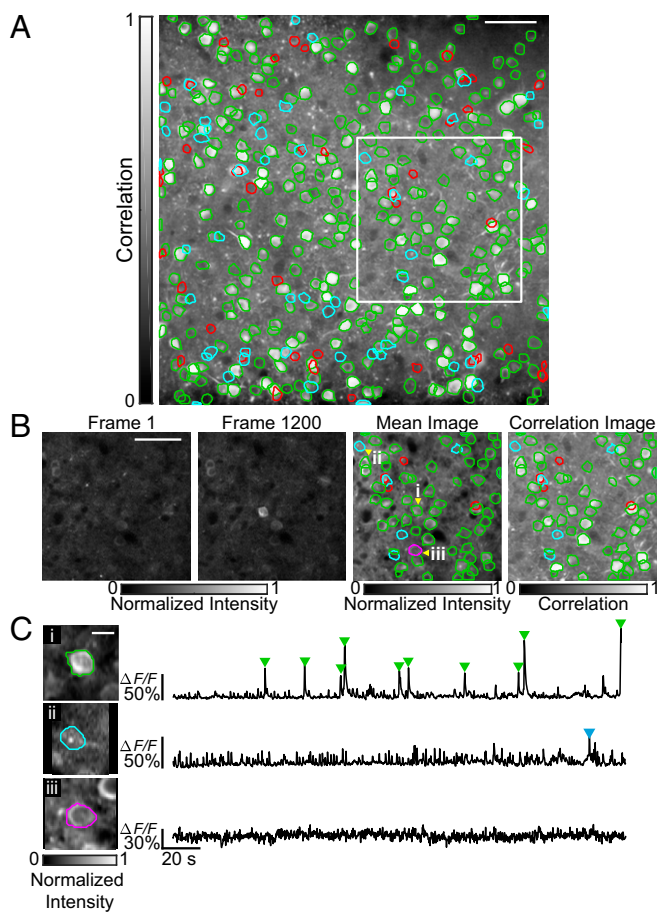


Fig. 3. STNeuroNet accurately identified active neurons from the ABO dataset. (A) Detection results from 1,200 frames (200 s) of a test video overlaid on the 200×200 pixels ($156 \mu\text{m} \times 156 \mu\text{m}$) cropped region from the correlation image of the data. The neuron detection metrics (recall, precision, F_1) for the whole-size data are (0.86, 0.88, 0.87). Green outlines, true positives; cyan outlines, false negatives; red outlines, false positives. (Scale bar, $50 \mu\text{m}$.) (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, $50 \mu\text{m}$.) (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, $10 \mu\text{m}$.)

B. The resulting distribution of mismatched cases (set of missed and falsely labeled neurons) was weighted toward neurons with low PSNR values, which challenge human perception during manual marking of the video (*SI Appendix, Fig. S4B*). Our framework achieved a higher F_1 score compared with grader 3 (mean of 0.84 versus 0.78, $P = 0.0013$; two-sided Wilcoxon rank sum test for $n = 10$ videos; *SI Appendix, Fig. S4C*). To mimic the case of semiautomatic marking, we asked a fourth grader to independently correct the ABO markings for these videos. Compared with grader 4, both grader 3 and STNeuroNet achieved lower F_1 scores ($P = 0.0002$ and 0.0036 , respectively; two-sided Wilcoxon rank sum test for $n = 10$ videos; *SI Appendix, Fig. S4C*), 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 10 videos as in the previous

section, from $275 \mu\text{m}$ below the pia in VISp. The neurons in these datasets were drawn from the 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 10 different mice, this time from a different cortical layer at $175 \mu\text{m}$ 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 $175 \mu\text{m}$ deep is putatively in layer 2/3, while the data from $275 \mu\text{m}$ 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. 5A). Quantitatively, the test set had bigger neurons (median of $112.6 \mu\text{m}^2$ versus $102.8 \mu\text{m}^2$; $P < 0.005$, two-sided Wilcoxon rank sum test over $n = 2,182$ and $3,016$ neurons, respectively; Fig. 5B) and lower densities of identified active neurons (0.0014 ± 0.0002 neurons per μm^2 versus 0.0019 ± 0.0003 neurons per μm^2 for 175 and $275 \mu\text{m}$ data, respectively; $P < 0.005$, 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 $275 \mu\text{m}$ data performed at indistinguishable levels on $275 \mu\text{m}$ test data and $175 \mu\text{m}$ data ($P = 0.1212$ for F_1 ; two-sided Wilcoxon rank sum test with $n = 10$ videos for both groups; Fig. 5C and *SI Appendix, Table S2*). Using the layer $275 \mu\text{m}$ data to set the algorithmic parameters of other methods, our framework achieved the highest mean F_1 score on the $175 \mu\text{m}$ data ($P < 0.005$, two-sided Wilcoxon rank sum test over $n = 10$ videos; Fig. 5D and *SI Appendix, Table S2*). Unlike our method, the F_1 scores of all other methods except UNet2DS were significantly lower on the $175 \mu\text{m}$ data compared with the $275 \mu\text{m}$ test data ($P = 0.006, 0.031, 0.021$, and 0.045 for CalmAn Online, CalmAn 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 labeled by different laboratories. 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. 6A and 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. 6B, *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 labeled the Neurofinder test set (*SI Appendix, Table S1*). Compared with the case of using Neurofinder's GT for training, the average F_1 score was not significantly different from the case of employing the markings from grader 1 for both training and testing ($P = 0.9372$, two-sided Wilcoxon rank sum test over $n = 6$ videos; Fig. 6C). Similar to the ABO dataset, we conducted an interhuman 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 an average

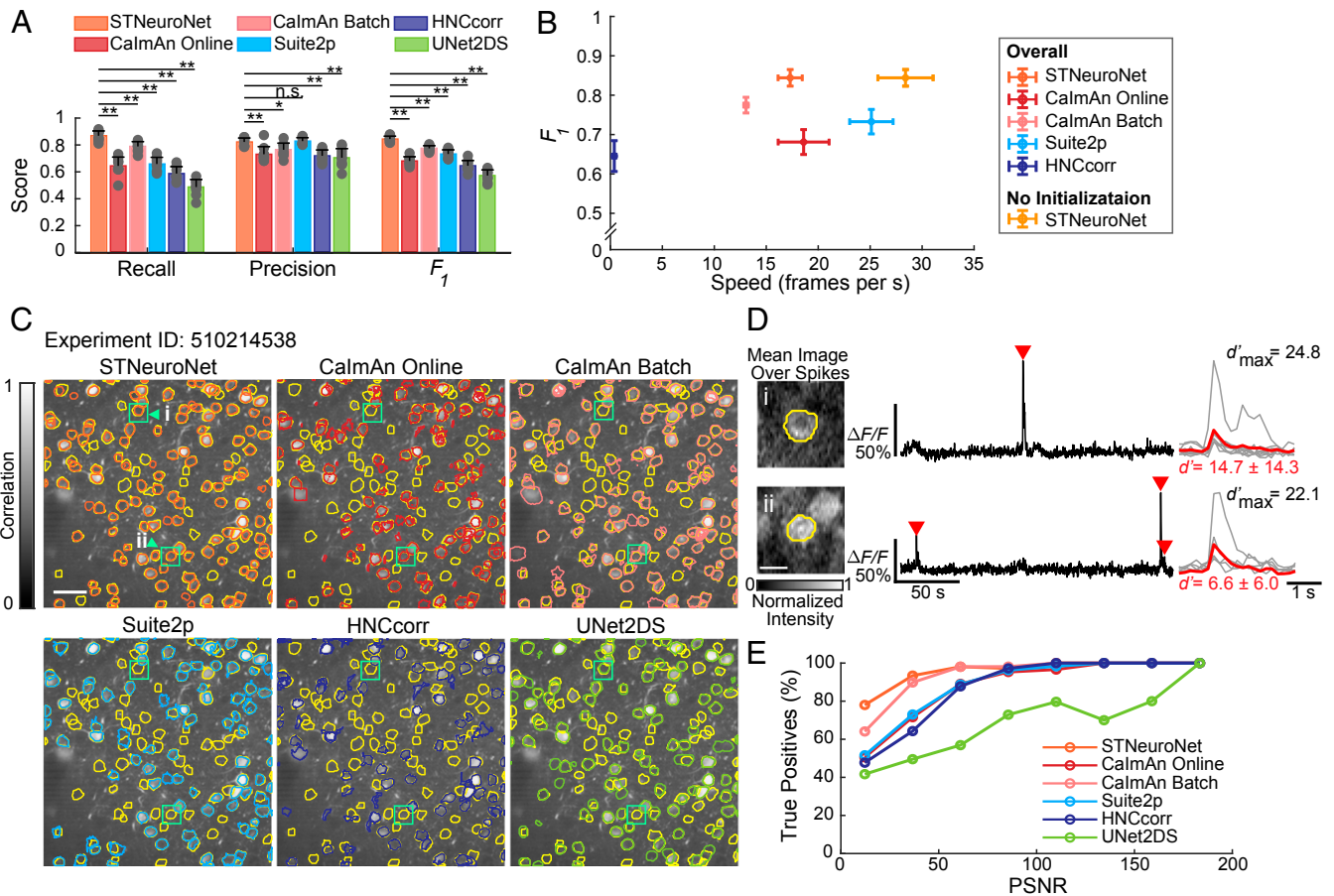


Fig. 4. STNeuroNet outperformed CalmAn, Suite2p, HNCcorr, and UNet2DS on the ABO dataset. (A) STNeuroNet's neuron identification score was superior to other methods ($*P < 0.05$ and $**P < 0.005$, two-sided Wilcoxon rank sum test, $n = 10$ videos; n.s., not significant), which was largely due to its 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 \pm SD of $F_1 = 0.57 \pm 0.04$ and speed = $2,263.3 \pm 2.6$ frames per s for $n = 10$ videos). Error bars in A and B are SDs for $n = 10$ videos. (C) Example data comparing the result of STNeuroNet to CalmAn (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, 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, 10 μm .) (E) Percentage of detected GT neurons versus binarized PSNR for all algorithms. The higher values of STNeuroNet's curve compared with other algorithms in the low PSNR regime indicate that our network identified a larger portion of neurons with low optical calcium response.

F_1 score comparable to that of grader 2 ($P = 0.2403$ 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. 6C).

Using our expert manual markings as GT for the Neurofinder dataset, we compared our framework to other methods (Fig. 6D 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 \pm SD of 0.70 ± 0.03 and 0.61 ± 0.08 , respectively; $P = 0.0649$, two-sided Wilcoxon rank sum test over $n = 6$ videos). Compared with all other methods, STNeuroNet's F_1 score was significantly higher ($P < 0.005$, 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 ($P = 0.485$, two-sided

Wilcoxon rank sum test for $n = 6$ videos), and the performance was comparable to that of Suite2p ($P = 1$, 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 with the network trained only on the Neurofinder training set ($P = 0.026$, 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 postprocessing 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 CalmAn, Suite2p, HNCcorr, UNet2DS, and an expert grader and generalized to segmenting active neurons from different cortical layers and regions

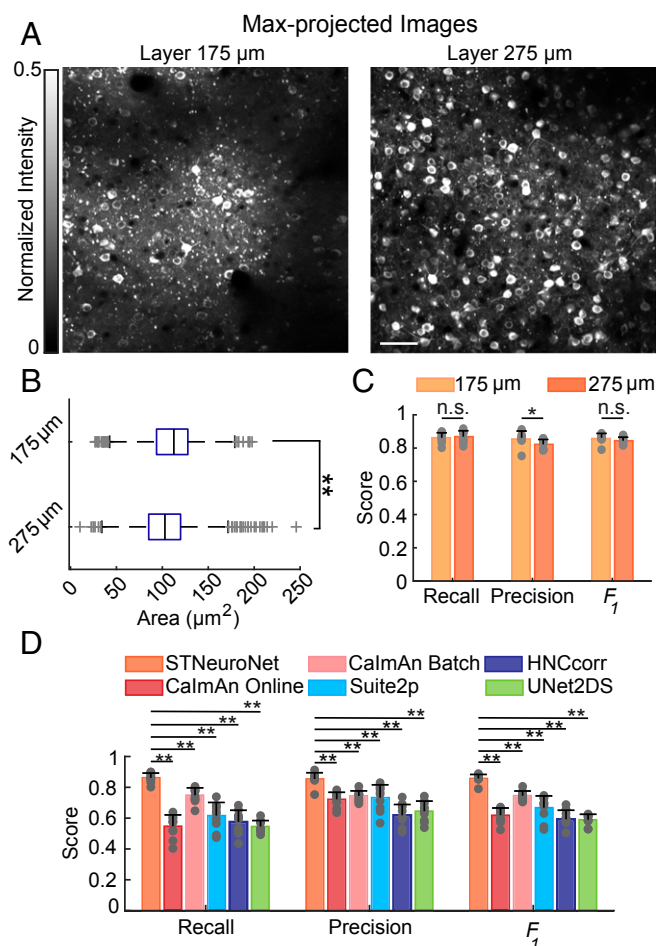


Fig. 5. Trained STNeuroNet performed equally well on data from a different cortical layer, outperforming CalmAn, Suite2p, HNCcorr, and UNet2DS. (A) Qualitative comparison between Layer 275 μm and 175 μm data from the ABO dataset. Images are the normalized maximum-value projection images over the entire recording of two sample data. (B) The area of active neurons labeled from the two cortical depths were different (** $P < 0.005$; $n = 2,182$ and 3,016 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 ($P = 0.5708$ and 0.1212 , respectively; * $P < 0.05$; $n = 10$ videos for both groups; n.s., not significant) 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 were superior compared with other methods (** $P < 0.005$; over $n = 10$ videos). All P values were derived using the two-sided Wilcoxon rank sum test.

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 (31), 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 spatiotemporal 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 with 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 with 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 spatiotemporal deconvolution methods of CaImAn and Suite2p, our proposed STNeuroNet extracts relevant spatiotemporal features from the imaging data without prior modeling; the deep learning approach could be more flexible for detecting arbitrary spatiotemporal features. Compared with 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 with 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, 32, 33) and fast spike detection algorithms (33–35), 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 nonoverlapping 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 (36), and batch normalization (37) 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 overtrained 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 setup improves performance of segmenting videos from a different experimental setup (SI Appendix, Table S5). These experiments confirm that other neuroscientists with significantly different recordings can take advantage of our trained network through transfer learning (38) 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

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 should extend the current framework to minimize parameter adjustments in pre- and postprocessing 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. The third and final stage is a set of postprocessing 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 to 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 nonuniform background illumination using homomorphic filtering (43) on each frame of the video. We formulated a high-pass filter by subtracting a low-pass Gaussian filter with SD of 0.04 μm^{-1} from 1. Then, we normalized the intensity of each video by dividing by its overall SD.

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. Three-dimensional convolutional layers extract local spatiotemporal information that capture the temporal dynamics of the input recording. We used the DenseVNet (31), implemented as part of NiftyNet (44), 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. 2B 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 (31, 45). 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 (31). All convolutional layers in DenseVNet perform 3D convolutions, use the rectified linear unit nonlinearity as the activation function, and consist of batch normalization (37) and dropout (36) 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 batchwise spatial dropout, as these did not have a significant effect on the final results reported in the original paper (31).

We made the following two modifications to DenseVNet for our application: (i) we changed the last convolutional layer of DenseVNet to have 10 output channels instead of the number of classes and (ii) we added a temporal max-pooling layer to the upsampled features, followed by a 2D convolutional layer with 10 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

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

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 reweighting (27).

Training Procedure and Data Augmentation. To create a large set of training samples, we cropped smaller windows of size $144 \times 144 \times 120$ voxels from the rotated (0° , 90° , and 180°) training videos and GT markings and applied random flips during training. We performed cropping using a spatiotemporal 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

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

where l is the 3D input sample to the network. We used the Adam optimizer (46) with learning rate of 0.0005 and minibatch 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).

Postprocessing 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 nonoverlapping 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 (47). We first calculated the distance transform image as the distance of each pixel to the nearest background pixel. We then applied the MATLAB (MathWorks) 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 to 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 ref. 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.

ABO 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 10 mice for comparison between algorithms and recordings at 175 μm deep in the cortex from a set of 10 different mice to assess the generalizability of all method.

The data were previously corrected for motion and had an accompanying set of automatically identified neurons (30). 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) 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 four-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: (i) The marked area had a bright ring with a dark center that changed brightness during the recording or (ii) the area was circular and had a size expected from a neuron (10 to 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 spatiotemporal 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 nonmatching 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 spatiotemporal neuron labels for training by extracting the neurons' active time intervals. We first separated traces of overlapping neurons using the linear regression approach of ref. 30. 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*). We considered neurons as active until 0.5 s after the detected spike times, equal to 3.5 times the half-decay time of spike-evoked GCaMP6f fluorescence signals reported in ref. 1.

Neurofinder Dataset and Labeling. The Neurofinder challenge consists of 19 training and 9 testing two-photon calcium imaging datasets acquired and annotated by four different laboratories. 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 semiautomatic method. Upon inspection of the fourth dataset (called the 03 set), we found that this dataset was labeled based on anatomical factors. We excluded the first and fourth sets from the comparison in *Results* 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 spatiotemporal labels for the Neurofinder training set by detecting neuronal spikes that satisfied the minimum required d' (*Spike Detection and Discriminability Index*), 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 (48, 49), 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 (49):

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

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 SD (σ) of the distribution of L under the hypothesis of a spike having occurred (49). In the above equation, $\Phi^{-1}(\cdot)$ is the inverse of the standard Gaussian cumulative distribution function (49).

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 (49). 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 (49). 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 (48):

$$(\bar{f}_s - \lambda)P_F = \lambda P_N, \quad [7]$$

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

In Eq. 7, \bar{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 $\lambda = 2.9$ spikes per s in accordance to previous experimentally obtained spike rates during similar behaviors (50). We then set a low $P_N = 0.035$, which corresponded to a spike detection threshold of $d' = 3.6$ based on Eqs. 7 and 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 with the ABO dataset.

Quantification of PSNR. To calculate the PSNR of neurons, we first separated traces of overlapping neurons using the linear regression approach of ref. 30. 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

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

where ΔF_{peak} is the difference between the biggest spike value and the baseline value, and σ_n is the noise SD calculated from nonactive 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:

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

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

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

These quantities derive from the number of manually labeled neurons (GT 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

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

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

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

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 Eq. 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 CalmAn Batch, we used all of the logical Cores of our central processing unit (CPU) (28 threads) for parallel processing. For STNeuroNet and CalmAn 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 CalmAn, Suite2p, HNCcorr, and the pre- and postprocessing 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 U-Net2D5 using a single NVIDIA GeForce GTX Titan X graphics processing unit (GPU). All CNNs in the CalmAn 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 \pm SD. We used two-sided Z-test for the statistical analysis of calcium transients' d' compared with 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; * $P < 0.05$; and ** $P < 0.005$. We determined results to be statistically significant when $P < 0.05$. 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 (28).

ACKNOWLEDGMENTS. We thank David Feng and Jerome Lecq 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 supported by National Institutes of Health Medical Imaging Training Program Pre-Doctoral Fellowships T32-EB001040 and P30-EY005722 and 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.

1. Chen T-W, et al. (2013) Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* 499:295–300.
2. Dombek DA, Harvey CD, Tian L, Looger LL, Tank DW (2010) Functional imaging of hippocampal place cells at cellular resolution during virtual navigation. *Nat Neurosci* 13:1433–1440.
3. Ghosh KK, et al. (2011) Miniaturized integration of a fluorescence microscope. *Nat Methods* 8:871–878.
4. Grienberger C, Konnerth A (2012) Imaging calcium in neurons. *Neuron* 73:862–885.
5. Yang W, Yuste R (2017) In vivo imaging of neural activity. *Nat Methods* 14:349–359.
6. Andilla FD, Hamprecht FA (2014) Sparse space-time deconvolution for calcium image analysis. *Advances in Neural Information Processing Systems 27*, eds Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (MIT Press, Cambridge, MA), Vol 1, 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 30*, eds Guyon I, et al. (Neural Information Processing Systems, San Diego), pp 2378–2388.
8. Guan J, et al. (2018) NeuroSeg: Automated cell detection and segmentation for in vivo two-photon Ca^{2+} imaging data. *Brain Struct Funct* 223:519–533.
9. Kaifosh P, Zaremba JD, Danielson NB, Losonczy A (2014) SIMA: Python software for analysis of dynamic fluorescence imaging data. *Front Neuroinform* 8:80.
10. Maruyama R, et al. (2014) Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Netw* 55:11–19.
11. Mukamel EA, Nimmerjahn A, Schnitzer MJ (2009) Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron* 63:747–760.
12. Pachitariu M, et al. (July 20, 2017) Suite2p: Beyond 10,000 neurons with standard two-photon microscopy. bioRxiv:10.1101/061507.
13. Pnevmatikakis EA, et al. (2016) Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron* 89:285–299.
14. Reynolds S, et al. (2017) ABLE: An activity-based level set segmentation algorithm for two-photon calcium imaging data. *eNeuro* 4:ENEURO.0012-17.2017.
15. Spaen Q, Hochbaum DS, Asin-Acha R (March 6, 2017) HNCcorr: A novel combinatorial approach for cell identification in calcium-imaging movies. 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. Athorpe N, et al. (2016) Automatic neuron detection in calcium imaging data using convolutional networks. *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Neural Information Processing Systems, San Diego), 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*, eds Cardoso MJ, et al. (Springer, Cham, Switzerland), pp 285–293.
19. Giovannucci A, et al. (2019) CalmAn 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* (IEEE, Piscataway, NJ), 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, Cham, Switzerland), pp 234–241.
23. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Piscataway, NJ), pp 4489–4497.
24. Varol G, Laptev I, Schmid C (2018) Long-term temporal convolutions for action recognition. *IEEE Trans Pattern Anal Mach Intell* 40:1510–1517.
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, Cham, Switzerland), pp 424–432.
26. Kamnitsas K, et al. (2017) Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med Image Anal* 36:61–78.
27. Milletari F, Navab N, Ahmadi S-A (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)* (IEEE, Piscataway, NJ), pp 565–571.
28. Soltanian-Zadeh S, Sahingur K, Blau S, Gong Y, Farsiu S (2019) Data from "STNeuroNet: An open-source convolutional neural network for research in calcium imaging analysis. GitHub." Available at <https://github.com/soltanianzadeh/STNeuroNet>. Deposited March 26, 2019.
29. Friedrich J, et al. (2017) Multi-scale approaches for high-speed imaging and analysis of large neural populations. *PLoS Comput Biol* 13:e1005685.
30. de Vries SE, et al. (June 29, 2018) A large-scale, standardized physiological survey reveals higher order coding throughout the mouse visual cortex. bioRxiv:10.1101/359513.
31. Gibson E, et al. (2018) Automatic multi-organ segmentation on abdominal CT with dense v-networks. *IEEE Trans Med Imaging* 37:1822–1834.
32. Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. *Neural Comput* 7:1129–1159.
33. Vogelstein JT, et al. (2010) Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J Neurophysiol* 104:3691–3704.
34. Deneux T, et al. (2016) Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nat Commun* 7:12190.
35. Friedrich J, Zhou P, Paninski L (2017) Fast online deconvolution of calcium imaging data. *PLoS Comput Biol* 13:e1005423.
36. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958.
37. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning* (International Machine Learning Society, Princeton), pp 448–456.
38. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? *Proceedings of the 27th International Conference on Neural Information Processing Systems* (MIT Press, Cambridge, MA), Vol 2, pp 3320–3328.
39. Rolnick D, Veit A, Belongie S, Shavit N (May 30, 2017) Deep learning is robust to massive label noise. arXiv:1705.10694.
40. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, Fergus R (June 9, 2014) Training convolutional networks with noisy labels. arXiv:1406.2080.
41. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (November 10, 2016) Understanding deep learning requires rethinking generalization. arXiv:1611.03530.
42. Pnevmatikakis EA, Giovannucci A (2017) NoRMCorr: An online algorithm for piecewise rigid motion correction of calcium imaging data. *J Neurosci Methods* 291:83–94.
43. Oppenheim AV, Schaffer R, Stockham T (1968) Nonlinear filtering of multiplied and convolved signals. *IEEE Trans Audio Electroacoust* 16:437–466.
44. Gibson E, et al. (2018) NiftyNet: A deep-learning platform for medical imaging. *Comput Methods Programs Biomed* 158:113–122.
45. Huang G, Liu Z, Weinberger KQ, van der Maaten L (2017) Densely connected convolutional networks. *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ), pp 2261–2269.
46. Kingma DP, Ba J (December 22, 2014) Adam: A method for stochastic optimization. arXiv:1412.6980.
47. Meyer F (1994) Topographic distance and watershed lines. *Signal Process* 38:113–125.
48. 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 Trans Biomed Eng* 65:2428–2439.
49. Wilt BA, Fitzgerald JE, Schnitzer MJ (2013) Photon shot noise limits on optical detection of neuronal spikes and estimation of spike timing. *Biophys J* 104:51–62.
50. Niell CM, Stryker MP (2010) Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron* 65:472–479.