Real-time Neuron Segmentation for Voltage Imaging

Yosuke Bando
Kioxia Corporation

Ramdas Pillai
Fixstars Solutions, Inc.

Atsushi Kajita
Fixstars Solutions, Inc.

Farhan Abdul Hakeem
Fixstars Solutions, Inc.

Yves Quemener
Fixstars Solutions, Inc.

Hua-an Tseng
Boston University

Kiryl D. Piatkevich
Westlake University

Changyang Linghu
University of Michigan

Xue Han
Boston University

Edward S. Boyden
MIT &
Howard Hughes Medical Institute

2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-3748-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/BIBM58861.2023.10385929

***Abstract*—In voltage imaging, where the membrane potentials of individual neurons are recorded at from hundreds to thousand frames per second using fluorescence microscopy, data processing presents a challenge. Even a fraction of a minute of recording with a limited image size yields gigabytes of video data consisting of tens of thousands of frames, which can be time-consuming to process. Moreover, millisecond-level short exposures lead to noisy video frames, obscuring neuron footprints especially in deep-brain samples where noisy signals are buried in background fluorescence. To address this challenge, we propose a fast neuron segmentation method able to detect multiple, potentially overlapping, spiking neurons from noisy video frames, and implement a data processing pipeline incorporating the proposed segmentation method along with GPU-accelerated motion correction. By testing on existing datasets as well as on new datasets we introduce, we show that our pipeline extracts neuron footprints that agree well with human annotation even from cluttered datasets, and demonstrate real-time processing of voltage imaging data on a single desktop computer for the first time.**

***Index Terms*—voltage imaging, real time, neuron segmentation, motion correction**

I. Introduction

Voltage imaging uses fluorescence microscopy to monitor neural activities of animals [1]. It uses fluorophores called *voltage indicators* that change their fluorescence depending on membrane potentials of neurons, providing signals that are close to neural voltage measured using more invasive, physically contacting devices such as patch clamps and electrodes. By capturing images at from hundreds to thousand frames per second (fps), voltage imaging enables temporally high-resolution detection of spikes as well as extraction of subthreshold activities, which is an advantage over more established Calcium imaging that uses Calcium ion concentration as a slow proxy for rapidly changing membrane potential [2].

However, voltage imaging presents a challenge in data processing. Even a fraction of a minute of recording with a limited image size yields gigabytes of video data consisting of tens of thousands of frames, and processing it to extract voltage traces from captured neurons can be time-consuming. Existing methods either require manual annotation [3]–[6] or spend significantly more time on image processing than the image recording time [7], [8], both of which can slow down iterative experiments by neuroscientists. Moreover, video frames captured with millisecond-level short exposures have a low signal-to-noise (SNR) ratio despite significant improvements in brightness and sensitivity of voltage indicators in recent years. Although the SNR of a voltage trace can be improved by combining observations from multiple pixels belonging to the same neuron, it can be challenging to identify (either manually or automatically) where neurons are in the video in the first place, if noisy signals are buried in background fluorescence especially in deep-brain samples. While background fluorescence may be reduced by two-photon microscopy [5], [6], [9] or optical techniques that steer light onto neurons of interest [10], [11], it is desirable to be able to use more prevalent, unmodified one-photon microscopes.

This paper presents a data processing pipeline that can segment footprints of spiking neurons from noisy voltage imaging data in real time, meaning that the runtime is equal to or shorter than the video recording time, on a single desktop computer. We build on the idea from previous work [8] that summarizes a video into a few still images. However, since we find there are cases where it is challenging to identify neurons given a few still images alone, our proposal is to split a video into time segments, and apply a summary image approach to them individually. We use the U-Net convolutional neural network (CNN) [12] to identify spiking neurons from summary images for each time segment, and aggregate them into a single set of ROI masks. This better exploits temporal information while still benefiting from reduced computation by summarization. In combination with GPU-accelerated motion correction we develop as a step before the segmentation, our pipeline as a whole runs in real time while still leaving some room for voltage trace extraction from ROIs.

We run our pipeline on existing datasets as well as on new datasets we introduce, and show that the processing times for all of the datasets are shorter than the respective video recording times. We also show that the ROI masks produced by our method have around 80% agreement (more precisely, an F_1 score of 0.8) with manual annotation on average.

In summary, the contributions of this paper are as follows.

- We propose a fast neuron segmentation method able to detect multiple, potentially overlapping, spiking neurons whose extracted footprints agree well with human annotation even for cluttered one-photon datasets.
- We implement a data processing pipeline incorporating the proposed segmentation method along with GPU-

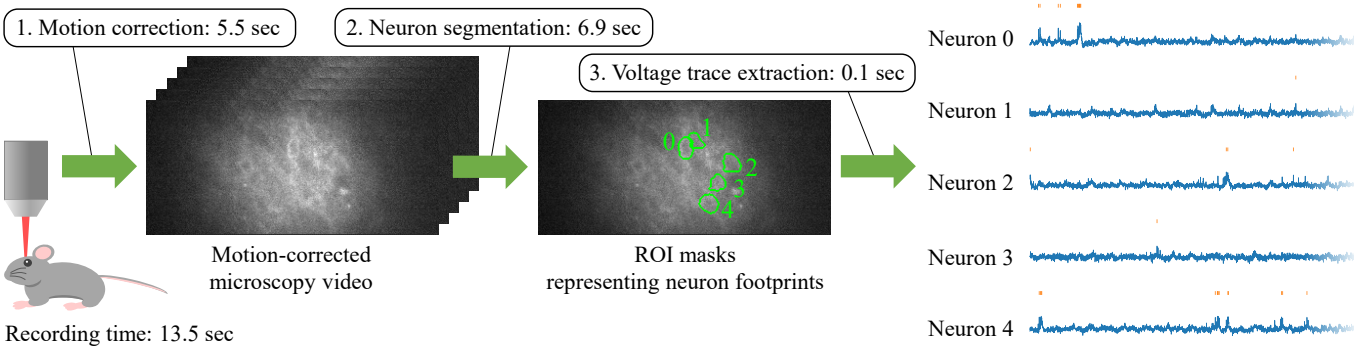


Fig. 1. Voltage imaging data processing pipeline. Our pipeline processes an input video in real time, meaning that the processing time is shorter than the recording time. As an example, given a video recorded in 13.5 sec capturing 10,000 frames at 741 fps, the three stages of the pipeline spend 5.5 sec, 6.9 sec, and 0.1 sec, respectively, totaling 12.5 sec, which is shorter than the recording time of 13.5 sec.

accelerated motion correction, and demonstrate real-time processing of voltage imaging data on a single desktop computer for the first time.

II. Pipeline Overview

Our pipeline consists of three stages as shown in Figure 1.

1) **Motion correction** for canceling motion so that the corrected video shows stationary neurons whose intensity variation comes from their changes in fluorescence.
2) **Neuron segmentation** for detecting neurons and delineating their contours from the background.
3) **Voltage trace extraction** to estimate the time-varying membrane potential of each segmented neuron.

The primary focus of this paper is the second stage of the pipeline: we propose a fast method for segmenting spiking neurons, which will be explained in Section III.

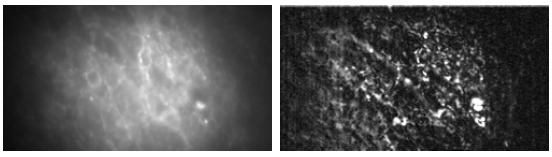
Here we briefly describe the other two stages. Our motion correction consists of carefully engineered implementations of well-known techniques. It calculates the zero-mean normalized cross-correlation (ZNCC) between images to measure their similarity. We tile 21x21-pixel patches to cover the image, and compute ZNCC for all the patches and candidate motion vectors in parallel on the GPU, while employing optimizations via table-based area sums [13]. Once patch-wise ZNCC values are computed, the GPU threads are synchronized and the values are aggregated to identify the most likely motion vector. Our voltage trace extraction takes the mean pixel intensity within each detected neuron ROI for each frame of the motion-corrected video. This is a rudimentary method for reference: more sophisticated alternatives may be used [3], [7], [8], [10].

III. Neuron Segmentation

This section presents details of the segmentation stage.

A. Motivation and Design

The previous methods for segmenting neurons from voltage imaging data take quite different approaches from each other. SGPMD-NMF [7] applies local nonnegative matrix factorization (NMF) to a video to decompose it into components each having spatially contiguous, temporally correlated pixels.



(a) Temporal average (b) Temporal correlation

Fig. 2. Summary images used in VolPy [8] for the input video in Figure 1.

Those components are considered neuron footprints. While this method has been shown to extract voltage signals including subthreshold dynamics with high fidelity, it takes time: using our desktop computer, it takes 33 min to analyze a 450x138-pixel video of 10,000 frames including the denoising time necessary before the local NMF (a similar runtime is reported in [7] on a computing cluster). Since the recording time of this video is 13.5 sec, the processing time is 148 times longer. SGPMD-NMF also requires some human intervention to identify blood vessels and initialize background regions.

In contrast, VolPy [8] is relatively fast and fully automatic. It summarizes an entire video into a few still images, which are input to the Mask R-CNN [14] in order to obtain neuron ROI masks. On top of the fact that it has to deal with only a few images, a CNN-based method is fast once trained, which makes a summary image method an appealing approach in terms of speed. While there are usually trade-offs between accuracy and speed, in this paper we are more interested in a speed-oriented solution that allows neuroscientists to iterate experiments quickly. Once good initial results are obtained, more elaborate analysis like SGPMD-NMF may be used later.

That being said, in order to deal with noisy one-photon voltage imaging data, we find summary image methods can be challenging. As an example, Figure 2 shows the temporal average and correlation images used by VolPy computed for the input video shown in Figure 1, which will be subsequently input to the Mask R-CNN. As can be seen, although the temporal average image provides much less noisy clues to neuron boundaries, the cluttered background makes it hard to delineate them. The temporal correlation image unfortunately provides

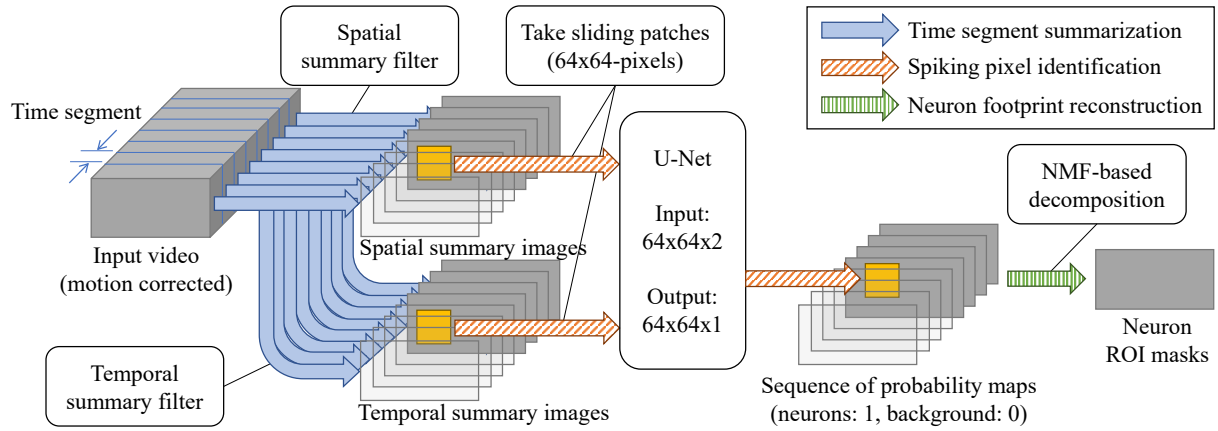


Fig. 3. Proposed segmentation subpipeline. The depth direction represents the time axis.

little information on where neurons are either, because the correlation between nearby pixels is buried in noise.

Thus, building on these previous works, we take a middle-ground approach where a video is split into time segments which are individually summarized and processed by a CNN as shown in Figure 3. This gives us a sequence of probability maps representing where neurons are likely to be spiking during each time segment, which is subsequently aggregated into ROI masks via NMF. This better exploits temporal information than immediately reducing the entire video into a few images, while still reducing computation by summarization.

B. Time Segment Summarization

We use two summary filters that each project a time segment along the time axis to produce a single summary image. In what follows, we denote the i-th time segment of the motion-corrected input video by $V_i(x,t)$, where x represents 2D spatial coordinates and $t \in [1, L]$ represents a frame number within the segment. We use a time segment length of $L = 50$ frames throughout this paper, which consistently produced good results for different datasets with varying frame rates.

The first filter is temporal average as

$$S_i(x) = \frac{1}{L}\sum_{t=1}^{L} V_i(x,t), \tag{1}$$

which produces a similar image to Figure 2(a), albeit with slightly increased noise due to a smaller number of frames to average. As it reveals spatial image features (i.e., boundaries) of neurons more clearly, we call it a *spatial summary image*.

The second filter is temporal maximum-minus-median (max-median for short) to enhance spiking neurons as

$$T_i(x) = \max_{t\in[1,L]}\{\bar{V}_i(x,t)\} - \operatorname*{median}_{t\in[1,L]}\{\bar{V}_i(x,t)\}, \tag{2}$$

where $\bar{V}_i$ is a spatially smoothed version of V_i using a Gaussian filter (we use a standard deviation of 3 pixels). If there is a spiking neuron at pixel x during this time segment, the temporal max operator identifies the peak of the spike while the temporal median extracts the baseline potential as

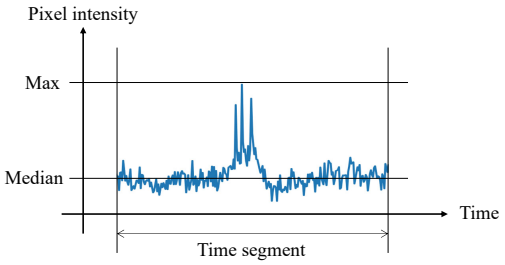


Fig. 4. Max-median filter.

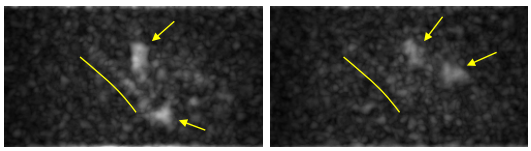


Fig. 5. Examples of temporal summary images using max-median filter.

shown in Figure 4, and therefore the difference gives a positive response. We find that pre-smoothing $\bar{V}_i$ is necessary to have a clear response. Figure 5 shows examples of the filtering results from two time segments of the input video shown in Figure 1, each showing two bright blobs (pointed to by the arrows) likely coming from spiking neurons. They additionally show some linear structures (the most visible one is indicated by the line next to it) from blood vessels as well as small blobs everywhere due to noise. Since these images reveal locations of temporal activities, we call them *temporal summary images*.

C. Spiking Pixel Identification

Based on the two summary images $S_i(x)$ and $T_i(x)$, we estimate where spiking neurons are likely to be during this (i-th) time segment. To this end, we employ the U-Net, a widely-used CNN for biomedical image segmentation [12]. As shown in Figure 6, we configure the U-Net in such a way that it takes as input 64x64-pixel patches from the two summary images, and outputs a 64x64-pixel image where each pixel represents the probability that there are spiking neurons at this

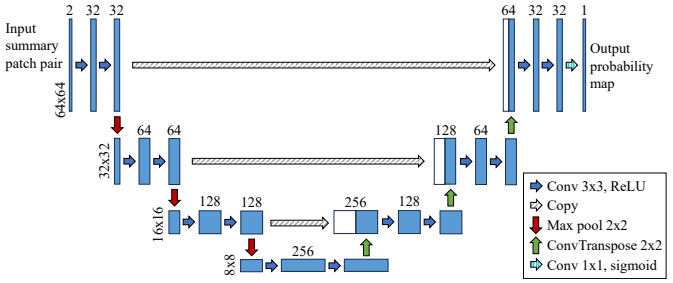


Fig. 6. Our lightweight U-Net configuration with a small input size of 64x64. The diagram convention follows the original U-Net work [12].

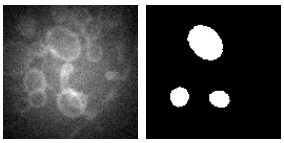


Fig. 7. Synthetic training data. An example video frame (left) and the corresponding binary mask indicating the footprints of spiking neurons (right).

Fig. 8. U-Net outputs representing the probability of each pixel belonging to spiking neurons. The input temporal summary images are shown in Figure 5.

pixel during this time segment. The patch-based approach is because we should be able to segment spatially distant neurons independently, and having a smaller input reduces the overall model size and makes the model easier to train.

To train the U-Net, we synthesize training datasets by voltage imaging simulation in order to be able to generate a larger number and variety of datasets than the real voltage imaging datasets we have, without having to annotate them. Figure 7 shows an example frame from a synthesized video and a binary mask indicating the locations of spiking neurons.

We synthesize 1,000 videos with varying configurations of neurons, blood vessels, illumination, and noise. Each video has 1,000 frames of 128x128 pixels, and after motion correction and summarization, we have 20 summary image pairs. For each summary image pair, we randomly pick 10 of 64x64-pixel patches (overlaps are allowed), resulting in 200 patches. Hence, in total we feed 200,000 patches to the U-Net for training, where 20% of them are used for validation. We use the binary cross-entropy loss and RMSProp optimizer.

We apply the trained U-Net to test data by taking sliding patches as shown in Figure 3, and merge U-Net outputs to reconstruct a single probability map for each summary image pair via weighted average. Output probability maps corresponding to the temporal summary inputs in Figure 5 are shown in Figure 8. Although some small blobs are incorrectly extracted, most noise and blood vessels are suppressed, and the locations of spiking neurons are delineated.

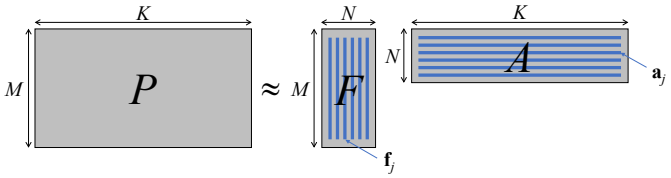


Fig. 9. NMF decomposition of a sequence of probability maps, represented as a matrix P, into neuron footprints F and their temporal activities A.

(a) Individual mask $b_i(x)$ (b) Aggregated mask $b(x)$

Fig. 10. Binary masks showing areas that potentially contain neuron footprints. (a) Mask from one time segment, corresponding to the left image of Figure 8. (b) Mask aggregating those from all of the time segments.

D. Neuron Footprint Reconstruction

Given a sequence of probability maps $p_i(x)$ indicating the likelihood of spiking neurons at pixel x during the i-th time segment, we decompose it into N neuron footprints and their temporal activity profiles by NMF as (see Figure 9):

$$P \approx FA. \tag{3}$$

Here, $p_i(x)$ is treated as a matrix $P \in \mathbb{R}^{M \times K}$, where M is the number of pixels per video frame and K is the number of time segments. The matrix $F \in \mathbb{R}^{M \times N}$ consists of N column vectors $\mathbf{f}_j \in \mathbb{R}^M$ representing the j-th neuron footprint, and the matrix $A \in \mathbb{R}^{N \times K}$ consists of N row vectors $\mathbf{a}_j \in \mathbb{R}^K$ representing the temporal profile of the j-th neuron.

In reality, applying NMF in Equation 3 directly does not produce good results because the U-Net outputs have some spurious detections, which can translate to false components. Moreover, NMF requires the number N of neurons as input, which is challenging to estimate. Therefore, we find the following procedure to be more reliable. First, we threshold the probability maps $p_i(x)$ to obtain binary masks $b_i(x)$, eliminating U-Net detections with small probabilities. We further eliminate regions whose shape is unlikely to be due to neurons by looking at their area, concaveness, and eccentricity. An example result of this process applied to the left image of Figure 8 is shown in Figure 10(a), where small blobs have been removed. After that, we project these cleaned-up masks along time segments by logically OR-ing them to obtain a single binary mask as $b(x) = \bigvee_{i=1}^{K} b_i(x)$, which indicates potential areas of neuron footprints as shown in Figure 10(b). Then, each connected component of $b(x)$ is where a few neurons might overlap (as visually noticeable at the top of Figure 10(b)), to which we apply NMF individually. By confining NMF to a small area, we can keep the matrix size as well as the potential number N of neurons small, making factorization more stable and faster. The bottom left image of Figure 1 shows reconstructed neuron footprints.

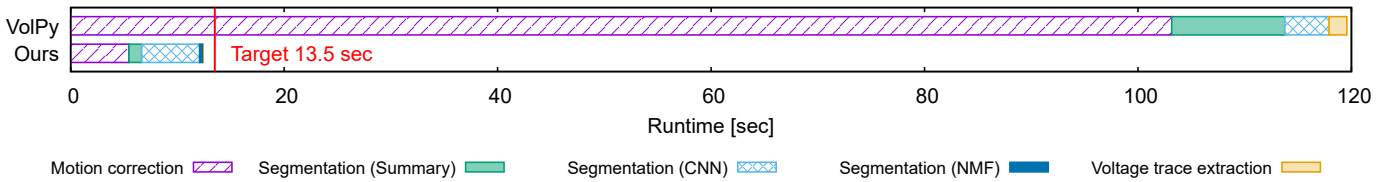


Fig. 11. Runtimes of VolPy [8] and our pipeline for one of HPC2 datasets shown in Figure 1. A shorter runtime is better. Our target is to process it within the video recording time of 13.5 sec. Our pipeline meets the target.

IV. Evaluation

We run our pipeline[1] on a single desktop computer as in Table I using two GPUs for motion correction and the U-Net.

TABLE I
Computational Environment

Component	Specification
CPU	AMD Ryzen Threadripper 3960X (24 cores)
RAM	DDR4 3200 MHz, 192 GB (6 ch. × 32 GB)
GPU	2 of NVIDIA GeForce RTX 2080 Ti
SSD	KIOXIA EXCERIA PRO 2 TB (PCIe Gen 4 ×4)
OS	Ubuntu 20.04.6 LTS, Linux kernel 5.15
Software	NVIDIA Driver 525.105.17, CUDA 12.0 Python 3.8, TensorFlow 2.4.1

Table II shows datasets we use. Each dataset group includes from 3 to 13 videos, totaling 37 videos. Each video has 10,000 through 20,000 frames. The first three dataset groups are curated and annotated by the VolPy authors [15]. Here we introduce a new dataset group[2], named "HPC2," consisting of 13 videos capturing mouse hippocampi using SomArchon voltage indicator [4]. While HPC uses patterned illumination to reduce background clutter [10], HPC2 uses normal one-photon wide-field microscopy. Hence, we believe HPC2 to be a useful addition as conventional microscopy data of deep-brain voltage imaging.

TABLE II
Datasets

Dataset group	Animal & brain region	Frame rate (fps)	Voltage indicator
L1 [15]	Mouse L1 cortex	400	Voltron [3]
TEG [15]	Zebrafish Tegmentum	300	Voltron [3]
HPC [15]	Mouse Hippocampus	1,000	paQuasAr3-s [10]
HPC2	Mouse Hippocampus	645-826	SomArchon [4]

A. Speed Evaluation

We begin by reporting the speed of our pipeline and that of VolPy for the example dataset from HPC2 shown in Figure 1. Figure 11 plots the runtimes of individual stages of each pipeline as well as a breakdown of the segmentation stage of each method into summary image generation, CNN (either U-Net or Mask R-CNN), and NMF (only used by our pipeline). Our target processing time is the recording time of this video, which is 13.5 sec. Our pipeline finishes processing within the target time, while VolPy's runtime is 8.9 times longer.

[1]Available at https://github.com/mitmedialab/voltage
[2]Available at https://zenodo.org/records/10020273

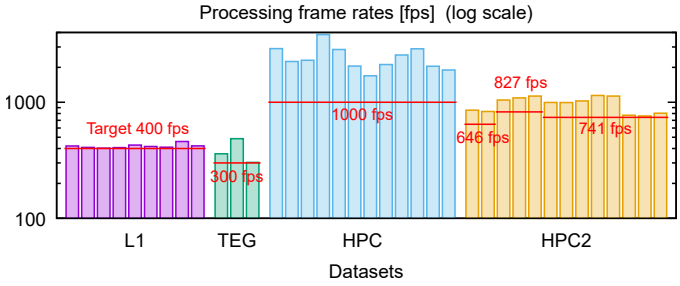


Fig. 12. Processing speeds of our pipeline in frame rates for individual datasets. Higher rates are better. Our target is to process a video at its recording frame rate. Our pipeline meets the target for all the datasets.

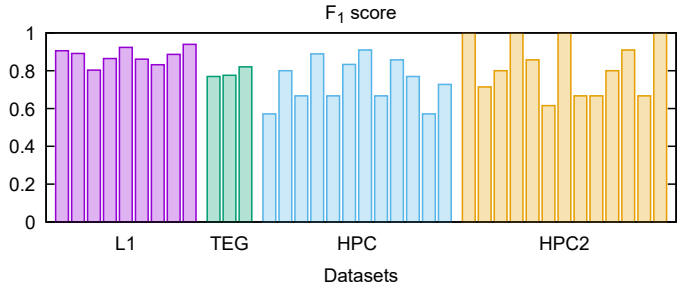


Fig. 13. F_1 scores of our neuron segmentation method for individual datasets. The possible range of scores is [0, 1]. Higher scores are better.

For all of the datasets, Volpy's runtime is longer than the video recording time (6.4, 7.8, 4.2, and 7.2 times longer on average for L1, TEG, HPC, and HPC2, respectively), whereas our pipeline achieves real-time processing. Figure 12 shows the processing speeds of our pipeline for individual datasets expressed in frame rates. For each dataset, its video recording frame rate is set as a target indicated by the horizontal bars. The processing speeds exceed the respective targets for all of the datasets, demonstrating real-time processing speeds.

B. Accuracy Evaluation

In order to assess segmentation accuracy, we rely on human annotation. While the agreement with human annotation by no means signifies whether a given method correctly detects true neurons, it represents how well it can replace manual labor that is routinely done in research [3]–[6]. We follow previous work [8], [16] and use the F_1 score as an accuracy metric at an intersection-over-union threshold of 0.3.

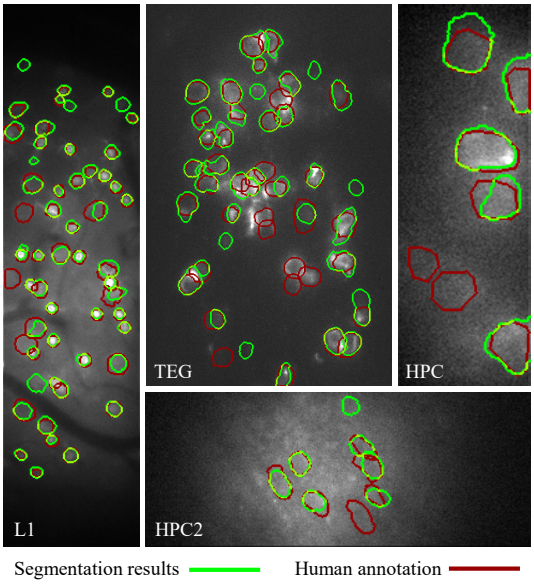


Fig. 14. Example results of our segmentation method.

Figure 13 shows the F_1 scores of our segmentation method for individual datasets. The scores range from around 0.6 to 1.0, and our method achieves an F_1 score of 0.8 on average. Figure 14 shows some of the segmentation results.

Table III shows the average F_1 scores (along with precision and recall) of VolPy and our method for each dataset group. For L1, TEG, and HPC, the VolPy scores are taken from their paper [8]. For HPC2, we evaluated VolPy through leave-one-out cross-validation. Namely, each one of the 13 videos was processed by the Mask R-CNN trained on the remaining 12 videos. Note that our U-Net was trained on synthesized data alone without using any of the test datasets. Table III indicates that VolPy and our method perform roughly equally well for cleaner datasets L1 and TEG. In contrast, HPC has more noise and background fluorescence even with patterned illumination, and HPC2 has even more cluttered backgrounds. Our method maintains high F_1 scores for these datasets.

TABLE III
PER-GROUP AVERAGE SEGMENTATION ACCURACY

Dataset	**VolPy**			**Ours**		
group	Prec.	Recall	F_1	Prec.	Recall	F_1
L1	**0.92**	**0.88**	**0.90**	0.91	0.85	0.88
TEG	0.78	0.74	0.76	**0.83**	**0.76**	**0.79**
HPC	0.61	0.77	0.66	**0.75**	0.77	**0.74**
HPC2	0.42	0.51	0.38	**0.89**	**0.79**	**0.82**

V. CONCLUSION

We have proposed a fast neuron segmentation method for voltage imaging, and demonstrated real-time processing on a single desktop computer for the first time. Future work includes experimenting with newer computer hardware and other CNN architectures, as well as incorporating better voltage trace extraction methods while accelerating them.

REFERENCES

[1] T. Knöpfel and C. Song, "Optical voltage imaging in neurons: moving from technology development to practical tool," *Nature Reviews Neuroscience*, vol. 20, pp. 719–727, 2019.
[2] R. Homma, B. J. Baker, L. Jin, O. Garaschuk, A. Konnerth, L. B. Cohen, and D. Zecevic, "Wide-field and two-photon imaging of brain activity with voltage- and calcium-sensitive dyes," *Phil. Trans. R. Soc. B*, vol. 364, pp. 2453–2467, 2009.
[3] A. S. Abdelfattah, T. Kawashima, A. Singh, O. Novak, H. Liu, Y. Shuai, Y.-C. Huang, L. Campagnola, S. C. Seeman, J. Yu, J. Zheng, J. B. Grimm, R. Patel, J. Friedrich, B. D. Mensh, L. Paninski, J. J. Macklin, G. J. Murphy, K. Podgorski, B.-J. Lin, T.-W. Chen, G. C. Turner, Z. Liu, M. Koyama, K. Svoboda, M. B. Ahrens, L. D. Lavis, and E. R. Schreiter, "Bright and photostable chemigenetic indicators for extended in vivo voltage imaging," *Science*, vol. 365, no. 6454, pp. 699–704, Aug. 2019.
[4] K. D. Piatkevich, S. Bensussen, H. Tseng, S. N. Shroff, V. G. Lopez-Huerta, D. Park, E. E. Jung, O. A. Shemesh, C. Straub, H. J. Gritton, M. F. Romano, E. Costa, B. L. Sabatini, Z. Fu, E. S. Boyden, and X. Han, "Population imaging of neural activity in awake behaving mice," *Nature*, vol. 574, pp. 413–417, 2019.
[5] Y. Bando, M. Wenzel, and R. Yuste, "Simultaneous two-photon imaging of action potentials and subthreshold inputs in vivo," *Nature Communications*, vol. 12, p. 2021, 7229.
[6] J. Platisa, X. Ye, A. M. Ahrens, C. Liu, I. A. Chen, I. G. Davison, L. Tian, V. A. Pieribone, and J. L. Chen, "High-speed low-light in vivo two-photon voltage imaging of large neuronal populations," *Nature Methods*, vol. 20, pp. 1095–1103, 2023.
[7] M. E. Xie, Y. Adam, L. Z. Fan, U. L. Böhm, I. Kinsella, D. Zhou, M. Rozsa, A. Singh, K. Svoboda, L. Paninski, and A. E. Cohen, "High-fidelity estimates of spikes and subthreshold waveforms from 1-photon voltage imaging in vivo," *Cell Reports*, vol. 35, no. 1, p. 108954, 2021.
[8] C. Cai, J. Friedrich, A. Singh, M. H. Eybposh, E. A. Pnevmatikakis, K. Podgorski, and A. Giovannucci, "VolPy: Automated and scalable analysis pipelines for voltage imaging datasets," *PLoS Comput Biol*, vol. 17, no. 4, p. e1008806, 2021.
[9] V. Villette, M. Chavarha, I. K. Dimov, J. Bradley, L. Pradhan, B. Mathieu, S. W. Evans, S. Chamberland, D. Shi, R. Yang, B. B. Kim, A. Ayon, A. Jalil, F. St-Pierre, M. J. Schnitzer, G. Bi, K. Toth, J. Ding, S. Dieudonne, and M. Z. Lin, "Ultrafast two-photon imaging of a high-gain voltage indicator in awake behaving mice," *Cell*, vol. 179, no. 7, pp. 1590–1608.e23, 2019.
[10] Y. Adam, J. J. Kim, S. Lou, Y. Zhao, M. E. Xie, D. Brinks, H. Wu, M. A. Mostajo-Radji, S. Kheifets, V. Parot, S. Chettih, K. J. Williams, B. Gmeiner, S. L. Farhi, L. Madisen, E. K. Buchanan, I. Kinsella, D. Zhou, L. Paninski, C. D. Harvey, H. Zeng, P. Arlotta, R. E. Campbell, and A. E. Cohen, "Voltage imaging and optogenetics reveal behaviour-dependent changes in hippocampal dynamics," *Nature*, vol. 569, pp. 413–417, 2019.
[11] S. Xiao, E. Lowet, H. J. Gritton, P. Fabris, Y. Wang, J. Sherman, R. A. Mount, H. Tseng, H.-Y. Man, C. Straub, K. D. Piatkevich, E. S. Boyden, J. Mertz, and X. Han, "Large-scale voltage imaging in behaving mice using targeted illumination," *iScience*, vol. 24, no. 11, p. 103263, 2021.
[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241.
[13] J. P. Lewis, "Fast template matching," in *Vision Interface*, Quebec City, Canada, May 1995, pp. 120–123.
[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.
[15] C. Cai, J. Friedrich, A. Singh, M. H. Eybposh, E. A. Pnevmatikakis, M. E. Xie, A. S. Abdelfattah, Y. Adam, E. R. Schreiter, A. E. Cohen, K. Podgorski, and A. Giovannucci, "VolPy: automated and scalable analysis pipelines for voltage imaging datasets," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4515768
[16] A. Giovannucci, J. Friedrich, P. Gunn, J. Kalfon, B. L. Brown, S. A. Koay, J. Taxidis, F. Najafi, J. L. Gauthier, P. Zhou, B. S. Khakh, D. W. Tank, D. B. Chklovskii, and E. A. Pnevmatikakis, "CaImAn an open source tool for scalable calcium imaging data analysis," *eLife*, vol. 8, p. e38173, 2019.