2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Open-Source Software for Real-time Calcium Imaging and Synchronized Neuron Firing Detection

Masaki Taniguchi[1], Taro Tezuka[2,*], Pablo Vergara[3], Sakthivel Srinivasan[3], Takuma Hosokawa[3], Yoan Chérasse[3], Toshie Naoi[3], Takeshi Sakurai[3], Masanori Sakaguchi[3]

Abstract— **We developed Carignan, a real-time calcium imaging software that can automatically detect activity patterns of neurons. Carignan can activate an external device when synchronized neural activity is detected in calcium imaging obtained by a one-photon (1p) miniscope. Combined with optogenetics, our software enables closed-loop experiments for investigating functions of specific types of neurons in the brain. In addition to making existing pattern detection algorithms run in real-time seamlessly, we developed a new classification module that distinguishes neurons from false-positives using deep learning. We used a combination of convolutional and recurrent neural networks to incorporate both spatial and temporal features in activity patterns. Our method performed better than existing neuron detection methods for false-positive neuron detection in terms of the F1 score. Using Carignan, experimenters can activate or suppress a group of neurons when specific neural activity is observed. Because the system uses a 1p miniscope, it can be used on the brain of a freely-moving animal, making it applicable to a wide range of experimental paradigms.**

I. INTRODUCTION

The development of calcium imaging and optogenetics enabled a new approach to brain science. By sending a signal to the brain when a specific neural activity pattern is detected, scientists can now test hypotheses about the functionality of activity patterns [1]. There have been several implementations of systems that enable such experiments, but in most cases, they either detect only simple patterns or use a two-photon (2p) miniscope. 2p miniscopes are much larger than one-photon (1p) ones and also require to constrain animals. This can make it difficult to observe the long-term behavior of animals. Some studies such as [2] use a 2p miniscope to observe neural activity during sleep. However, the large size of the 2p miniscope is possibly hindering natural sleeping behaviors.

For behavior studies, 1p miniscopes are indispensable because of their small size. However, they have lower resolution and have difficulty in identifying neurons. To retrieve neural activity at high precision, they need to be combined with a sophisticated machine learning method. One approach to solving this problem is CaImAn [3], a widely-used open-source library that processes calcium imaging data. It extracts neural activity using constrained nonnegative matrix factorization (CNMF-E) [4]. We extend this approach further by developing an open-source real-time system that triggers an external device when a specific neural firing pattern is detected in the calcium imaging video. Such a system, together with optogenetics, enables closed-loop experiments where specific types of neurons are excited or inhibited when a firing pattern is observed.

One example that signifies the power of closed-loop experiments was conducted by Gridchyn et al. [5]. Using tetrodes and targeting CA1 neurons in the mouse hippocampus, they disrupted the reactivation of learning-related ensembles occurring after learning, which led to memory impairments. They used an online decoding method developed by Ciliberti et al. [6]. Grosenick et al. provided a good survey on closed-loop approaches, including ones using tetrodes, 2p, and 1p imaging [7].

Ghandour et al. showed that engram cells display repetitive ensemble activity after learning [8]. The function of ensemble reactivation can be potentially addressed with a closed-loop experiments.

To enable such experiments based on calcium imaging, we developed **Carignan**, a real-time calcium imaging software that automatically extracts synchronized activity of neurons and triggers an external device. Our system retrieves video streams from a 1p miniscope and detects synchronous neural activity patterns in real-time. When a pattern is detected, the system triggers a laser device, enabling a feedback loop. Using our system, experimenters can activate or suppress a group of neurons when specific neural activity is observed. Because the system uses a 1p miniscope, it can be used on the brain of a freely-moving animal, making it applicable to a wide range of experimental paradigms, such as in [9].

The main contributions of our work are as follows.

1) We developed Carignan, a fully open-source image processing software for real-time processing of calcium imaging data obtained from 1p or 2p miniscopes.
2) A new neuron extraction module that classifies neurons from background noise using a combination of convolutional and recurrent neural networks.
3) Real-time control device that is activated by neuron firing patterns.

II. RELATED WORK

A. One-photon imaging

There are currently two types of miniscopes used for observing neural activity; one-photon (1p) and two-photon

[1]Graduate School of Library, Information and Media Studies, University of Tsukuba, Tsukuba, Japan
[2]Faculty of Library, Information and Media Science / Center for Artificial Intelligence Research (C-AIR), University of Tsukuba, Tsukuba, Japan
[3]International Institute for Integrative Sleep Medicine (WPI-IIIS), University of Tsukuba, Tsukuba, Japan
*Corresponding author. tezuka@slis.tsukuba.ac.jp

(2p). 2p miniscopes are larger in size and more expensive. 1p miniscopes are much smaller and less costly, but images obtained from 1p miniscopes tend to be unclear and are largely affected by background noise. Therefore, it is a common practice to process the images using a statistical algorithm for data cleansing. One example is CNMF-E developed by Zhou et al.[4] which extends a method using CNMF proposed by Pnevmatikakis et al.[10]. CNMF-E decomposes video into spatial and temporal components using non-negative matrix factorization. However, the output of CNMF-E usually contains many false-positives. For example, blood vessels are often classified as neurons.

There have been many studies that attempt to address this problem [11]. OnACID uses space correlation to remove false-positives [12]. CaImAn is an open-source Python implementation that uses a convolutional neural network (CNN) to identify false-positives [3]. CaImAn processes each frame separately and does not use the temporal dynamics of retrieved patterns. This approach is sufficient for images from a 2p miniscope. However, images from 1p miniscopes contain more noise, so the temporal dynamics need to be incorporated to achieve high precision. Tran et al. [13] did so by concatenating spatial footprints and temporal traces into feature vectors and inputting them into a conventional auto-ML method such as Auto-Sklearn [14] and TPOT [15]. We extend their approach by utilizing the temporal information more effectively using deep learning.

B. Image processing of calcium imaging data

There are many implementations of batch calcium imaging analysis [16], [17], [18], [19]. Programs that process in real-time have gradually become commonly used. Lee et al. implemented a system based on a data-flow framework that enables the replacement of modules [20]. Since their module for spatial component extraction was based on SimpleBlobDetector in OpenCV, it is not effective enough for unclear images obtained from a 1p miniscope. The PCA/ICA-based method [21] proposed by Mukamel et al. is used in analysis software by Inscopix, Inc. However, since this analysis software is not open-source software, it was unsuitable for use in combination with external modules. In addition, this method did not have any special consideration for 1-photon images, which was inappropriate for our purposes.

Most batch CNMF algorithms are computationally intensive, but Friedrich et al. developed an efficient algorithm that can run in real-time [22], [23] using sparse non-negative deconvolution [24], [10]. CaImAn uses this algorithm, but its latest version could only process already-recorded video data. By extending CaImAn, we developed Carignan, an integrated system that processes calcium imaging data in real-time and triggers an external device when the system detected a specific neural activity pattern.

III. PROPOSED SYSTEM

Carignan's system structure is summarized in Figure 1. We describe each part of the whole system in detail.

A. Miniscope

We used the UCLA-Miniscope V3 as our default 1p miniscope because, to the best of our knowledge, it is the only miniscope that is compatible with the USB Video class. This compatibility enables a smooth acquisition of a video stream from our application program implemented in Python. The user can adjust the gain, field-of-view (FOV), and frames-per-second (FPS) on a graphical user interface (GUI) included in our system. The system uses the VideoCapture class in OpenCV so that it can process already-recorded video as well.

Figure 2 shows an overview of processing calcium imaging data using our system.

B. Signal processing

This section describes the overall system. After setting up the aforementioned microscope, the system automatically acquires video frames at a set FPS and starts online CNMF-E analysis. During the frame-by-frame analysis, the system detects neural activity in the video captured by the microscope and can control external devices by triggering specific firing patterns. To specify the firing pattern, we need to record twice during the experiment as shown in Figure 1-A. During the first recording, we capture a video of a particular length and output the neurons' spatial component and temporal trace. If a characteristic firing pattern is found in the spatial component and temporal trace, the user creates a pattern file, and the recording is done again. During the second recording, the user can perform a number of operations on the external device at times when neural activity corresponding to the pattern file occurs. In the following sections, we will describe the individual modules of online CNMF-E in detail as shown in Figure 1-B.

1) Initialization: We set the length of the video segment used for matrix decomposition to 500 frames. The number of frames must be more than 500 to detect false-positives. After performing motion correction, bare initialization [3] is used in our system's initialization stage. This algorithm is used to estimate the background by performing CNMF repeatedly in short batches. The neurons detected in this stage are also selected through the false-positive detection module described below.

2) Motion correction: This system can use the same motion correction method and the same parameters as in the CaImAn implementation. However, since this system mainly handles 1p images, we set the gSig_filt parameter and apply a high-pass filter to eliminate the background's effect as much as possible.

3) Source extraction: We use the OnACID implementation in CaImAn, which enables us to use the same parameters. However, since we are dealing with 1p images, parameters such as gSig, min_corr and min_pnr need to be carefully considered. Also, during the second recording, it will not find a new spatial footprint to maintain consistency with the pattern file.

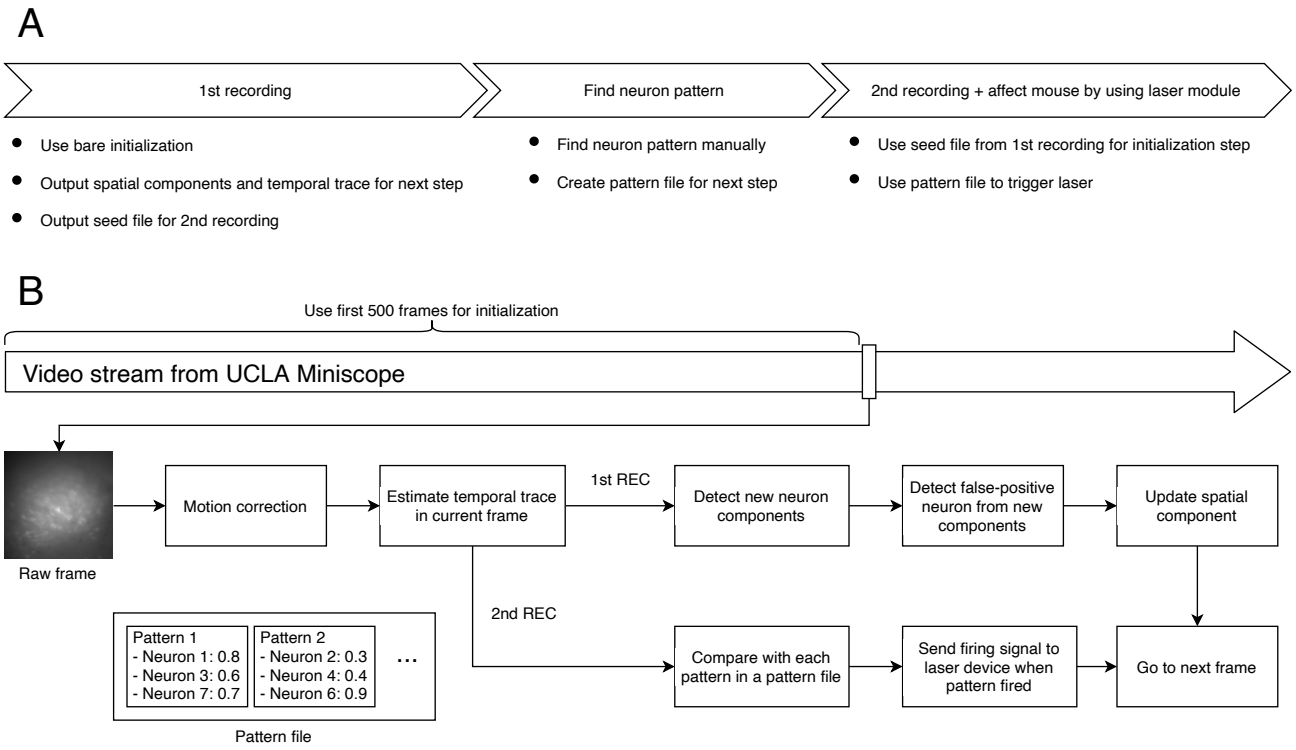

Fig. 1. Overall workflow of our system. The first 500 frames are used for initialization. For each time frame after that, neuron candidates are extracted and then filtered using the false-positive detection module.

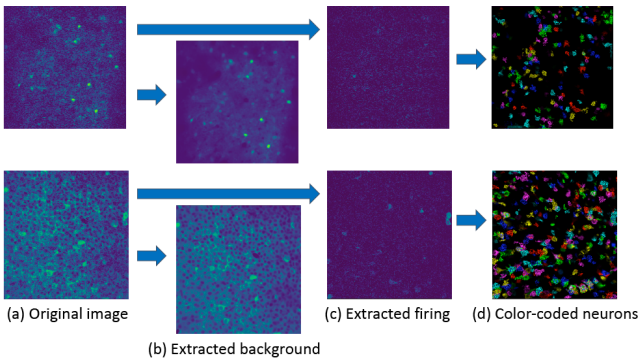

Fig. 2. Calcium imaging data processed using our proposed method. Each row indicates an example. The raw image is obtained from a miniscope (a). Its background is extracted (b). The background may contain artifacts and neurons that did not fire frequently enough. The firing of neurons is also extracted (c). Color coding of extracted neurons shows that we can observe numerous neurons (d).

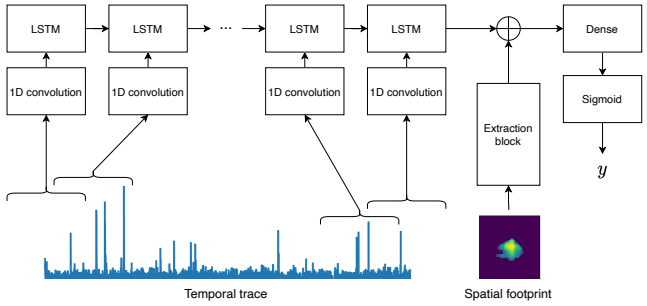

Fig. 3. Architecture of the false-positive detection module.

4) False-positive detection: We proposed to use deep learning to remove false-positives from the candidate neurons obtained by CNMF-E. Namely, we combined a CNN with a recurrent neural network. The TPOT model of cnmfe-reviewer [13] is also implemented in this application, and users can choose to use it.

5) Matching with manually defined patterns: In the second recording, the pattern file provided by the user contains the IDs of the neurons and the thresholds for detecting their firings in the form of a matrix. The pattern file can contain multiple firing patterns. The system can detect a neuron's activity in each frame and send an arbitrary signal to any USB signaling device at any time length when all the thresholds for all neurons of a pattern in the pattern file are exceeded. The system can connect to an external device that uses an Arduino microcontroller.

6) Visualization: We implemented a visualization module that enables the user to see the analysis results in real-time during the recording experiment. Figure 4 is a capture of the actual software. The left side shows the raw video being recorded. The four images on the right from the upper left to lower right are the colored motion corrected raw video, the separated background, all extracted neurons, and each extracted neuron colored differently, respectively. The seek bar at the bottom of the screen enables the user to adjust the dynamic range of the visualized videos.

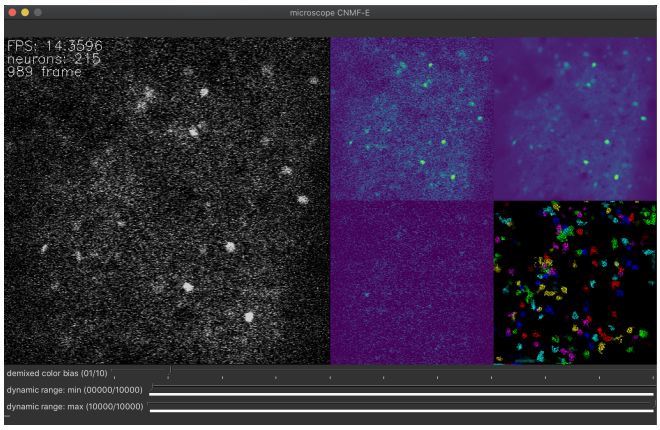

Fig. 4. Real-time plotting of neural activity using the graphical user interface (GUI) of Carignan. The left panel shows the raw Calcium imaging, and the small panels on the right show the processed results.

C. Code availability

The entire code of the system is available as an open-source software[1]. The repository shows a link to sample data and procedures for running the software.

IV. EXPERIMENTS

A. False-positive detection

We evaluated our proposed method for false-positive detection using annotated data. All animal experiments for creating the datasets were approved by the University of Tsukuba Institutional Animal Care and Use Committee. The detailed architecture of our method is shown in Figure 3. For each neuron detected by CNMF-E, the temporal trace is sent to the long short-term memory (LSTM) after passing through a one-dimensional convolutional layer. The spatial footprint is sent to the extraction module, which is transformed into a one-dimensional vector and concatenated with the output of the LSTM. Finally, the predictions are obtained after the fully-connected layers. In our experiment, 500 frames are used for a temporal trace, and an 80×80 pixel image is used for a spatial footprint to match the conditions of cnmfe-reviewer [13].

We tested four methods proposed in cnmf-reviewer [13] and several deep learning-based methods using the same test dataset. We evaluated our system in terms of the scores and prediction speed. We trained the decision tree, k-nearest neighbors, and all the deep learning models using the training data in cnmfe-reviewer.

In Table I, the top rows show the results of the cnmfe-reviewer methods. The middle rows show those using a simple deep learning model without a pre-trained extractor. The bottom row shows those using the feature extraction part of the pre-trained classification models in the torchvision library. The decision tree, k-nearest neighbors, TPOT, and AutoSklearn all use temporal trace and spatial footprints. The models that exceed the performance of the TPOT model, which is the best performing model in cnmfe-reviewer, are written as bold text under "Accuracy" and "F1". We measured the processing rate using a MacBook Pro 2017 (Intel Core i5, 16GB RAM). Models that reached 30 Hz or faster are indicated as bold text under "Processing Rate".

Table I shows that the k-nearest neighbor and Auto-Sklearn of cnmfe-reviewer are not practical for real-time applications in terms of computation time. The small deep models without pre-training show that temporal trace and spatial footprint contributed to classification performance. Temporal trace alone (LSTM or Conv-LSTM) and spatial footprint alone (2D-CNN) could not outperform TPOT, but the combined method (Conv-LSTM + 2D-CNN) did. The same trend is also shown in the results for the cnmfe-reviewer and deep learning models. Furthermore, many classifiers with torchvision pre-trained extractors show that the accuracy does not improve much even when using a huge image classification model pre-trained by a large-scale dataset. One possibility that the amount of data used for training is insufficient for the massive number of parameters in the model. Another possibility is that the ImageNet dataset used for pre-training is not suitable for retrieving identified neurons. Also, such large-scale models are generally not practical for real-time applications in terms of speed. On the basis of the experimental results, we implemented two types of model in the feature extraction module. One is a simple 2D-CNN without transfer learning. The other is a ShuffleNet V2 [25] with transfer learning. The detailed architecture of the 2D-CNN model is shown in Table II.

B. Overall processing speed

To measure Carignan's overall performance, we measured the computation time using two public datasets and two private videos taken by our group. The results are summarized in Table III. The two public datasets are both 2p images included in the neurofinder dataset. The non-public data are both 1p images taken by our group. Video-1 and Video-2 were taken using the nVoke miniscope and UCLA-Miniscope, respectively.

V. DISCUSSION

From the experiments described in the previous section, we found that our deep learning model performed better for detecting false-positive neurons in terms of accuracy. TPOT and other simple machine learning models were faster, but when we measured the entire system's computation time, the processing rate was higher for our deep learning model. This is due to the difference in the number of detected neurons, not that in the processing speed of classification. Table IV shows the number of neurons detected from each video, the number of neurons accepted or rejected by the detector, and the acceptance rate. All of the models for false-positive neuron detection used in this validation have been trained on the dataset provided by Tran et al. [13]. This dataset consists of hippocampal CA1 neurons captured by a 1p microscope from multiple mice. On the other hand, the neurofinder data used in this study were both taken with a 2p microscope, and for data 01.00, the images are of V1 neurons. This suggests

[1] https://github.com/tzklab/carignan

TABLE I

PERFORMANCE COMPARISON OF FALSE-POSITIVE NEURON DETECTION.

Method	Accuracy	F1	Precision	Recall	Processing Rate (Hz)
decision tree	0.825	0.861	0.865	0.856	**6852.094**
k-nearest neighbors	0.844	0.882	0.843	0.926	7.126
TPOT	**0.878**	**0.910**	0.854	0.975	**10523.117**
AutoSklearn	0.877	0.907	0.867	0.951	2.066
small deep models without pretrain					
LSTM (Only temporal trace)	0.849	0.888	0.840	0.942	**814.352**
Conv-LSTM (Only temporal trace)	0.858	0.896	0.837	0.964	**74.815**
2D-CNN (Only spatial footprint)	0.884	0.909	0.905	0.912	**107.711**
Conv-LSTM + 2D-CNN	**0.894**	**0.920**	0.882	0.961	**42.631**
with torchvision pretrained extractors					
Conv-LSTM + Alexnet	**0.882**	0.905	0.928	0.883	**54.143**
Conv-LSTM + VGG11	0.873	0.894	0.945	0.849	15.978
Conv-LSTM + VGG13	0.877	0.899	0.940	0.861	13.201
Conv-LSTM + VGG16	**0.895**	**0.918**	0.903	0.934	10.330
Conv-LSTM + VGG19	**0.891**	**0.913**	0.925	0.900	8.570
Conv-LSTM + Squeezenet	0.633	0.775	0.633	1.000	**45.342**
Conv-LSTM + Densenet 121	**0.897**	**0.919**	0.913	0.925	12.264
Conv-LSTM + Densenet 161	**0.897**	**0.919**	0.918	0.919	6.572
Conv-LSTM + Densenet 169	**0.890**	**0.912**	0.929	0.895	9.597
Conv-LSTM + Densenet 201	**0.893**	**0.916**	0.915	0.917	7.381
Conv-LSTM + Mobilenet V2	**0.898**	**0.923**	0.885	0.965	28.414
Conv-LSTM + ResNet 18	**0.894**	**0.915**	0.925	0.906	28.629
Conv-LSTM + ResNet 34	**0.895**	**0.916**	0.928	0.904	18.625
Conv-LSTM + ResNet 50	**0.893**	**0.914**	0.927	0.902	13.350
Conv-LSTM + ResNet 101	**0.897**	**0.917**	0.926	0.909	8.809
Conv-LSTM + ResNet 152	**0.898**	**0.919**	0.915	0.924	6.729
Conv-LSTM + ResNext 50	**0.897**	**0.918**	0.928	0.907	12.013
Conv-LSTM + ResNext 101	**0.887**	0.908	0.936	0.882	5.117
Conv-LSTM + WideResNet 50	**0.903**	**0.923**	0.930	0.915	6.503
Conv-LSTM + WideResNet 101	**0.893**	**0.915**	0.927	0.902	3.745
Conv-LSTM + ShuffleNet V2	**0.890**	**0.914**	0.904	0.925	**32.052**
Conv-LSTM + GoogleNet	**0.891**	**0.912**	0.932	0.893	22.400

TABLE II

ARCHITECTURE OF THE 2D-CNN MODEL

Module	Parameters
input	height: 80, width: 80, channel: 1
2D Conv + BN + ReLU	in: 1ch, out: 32ch, kernel size: 1, stride: 1
Maxpool	kernel size: 2, stride: 2
2D Conv + BN + ReLU	in: 32ch, out: 32ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 32ch, out: 8ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 8ch, out: 8ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 8ch, out: 32ch, kernel size: 1, stride: 1
Maxpool	kernel size: 2, stride: 2
2D Conv + BN + ReLU	in: 32ch, out: 64ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 64ch, out: 16ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 16ch, out: 16ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 16ch, out: 64ch, kernel size: 1, stride: 1
Maxpool	kernel size: 2, stride: 2
2D Conv + BN + ReLU	in: 64ch, out: 128ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 128ch, out: 32ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 32ch, out: 32ch, kernel size: 1, stride: 1
2D Conv + BN + ReLU	in: 32ch, out: 128ch, kernel size: 1, stride: 1

that the difference in the number of neurons detected by each model shown in Table IV is due to each model's performance in terms of how robust it is against the difference in the type of neurons and the camera environment. Considering that the training dataset's false neuron ratio is 15%, the two deep learning models are deemed to have lower robustness against environmental changes than the TPOT model. It is recommended that the user trains the deep learning models using images of neurons in the same domain as those used for pattern detection. Also, the acceptance rate dropped for all models in the 01.00 data, which recorded different types of neurons. It suggests that neurons' essential features can be learned even when using pseudo-false data created by adding mechanical deformations to the true data, as for this dataset.

VI. CONCLUSION

We developed Carignan, a real-time image processing application that (1) retrieves calcium imaging video from a one-photon (1p) miniscope, (2) finds synchronized neural activity patterns by machine learning, and (3) sends signals to an external device when a specific pattern is observed. In addition to making existing algorithms run in real-time, we developed a new classification module that distinguishes neurons from false-positives using deep learning. We show through experiments that the module achieves a higher precision than existing methods.

In future work, we plan to conduct in-vivo experiments using our system. We expect our system will uncover functions of neural ensembles in the brain at a scale that has not been achieved before. In our current system, synchronized firing patterns are constructed manually. We will consider using unsupervised machine learning methods to add a mechanism that automatically finds firing patterns. We expect such an extension will contribute to the scientific investigation of neural ensemble activities and their functions. Since the learning of firing patterns is not a part of our system, it needs to be carried out in advance. Because a single experiment is

TABLE III

COMPARISON OF COMPUTATION TIME

Video data	TPOT	CNN	ShufflenetV2
neurofinder 01.00 Neuron number: 345 Camera: 2p Recording rate: 7.5 Hz Resolution: 512×512 bits per pixel: 16 bit	16.156 Hz	17.709 Hz	18.329 Hz
neurofinder 03.00 Neuron number: 621 Camera: 2p Recording rate: 7.5 Hz Resolution: 498×490 bits per pixel: 16 bit	15.235 Hz	16.580 Hz	17.432 Hz
Video 1 Camera: 1p (nVoke) Recording rate:10 Hz Resolution: 321×213 bits per pixel: 16 bit	44.647 Hz	46.645 Hz	44.603 Hz
Video 2 Camera: 1p (UCLA-Miniscope) Recording rate: 10 Hz Resolution: 480×480 bits per pixel: 8 bit	7.511 Hz	16.501 Hz	13.561 Hz

TABLE IV

ACCEPTANCE RATES OF FALSE-POSITIVE NEURON DETECTION MODULES IN THE OVERALL SYSTEM

Method	accept	reject	acceptance rate
neurofinder 01.00 (345 neurons)			
TPOT	323	49	0.868
Conv-LSTM + 2D-CNN	13	508	0.025
Conv-LSTM + ShuffleNet V2	121	335	0.265
neurofinder 03.00 (621 neurons)			
TPOT	548	13	0.977
Conv-LSTM + 2D-CNN	21	612	0.033
Conv-LSTM + ShuffleNet V2	198	397	0.333

divided into two separate steps, it is prone to errors such as an unintended changes in camera position. We would like to reduce such a risk by connecting the learning phase seamlessly to the current system such that experiments can be carried out in a single shot.

ACKNOWLEDGMENTS

This work was partially supported by grants from the World Premier International Research Center Initiative from MEXT; JST CREST Grant JPMJCR1655; JSPS KAKENHI Grants 16K18359, 15F15408, 26115502, 25116530, 16H06280, 19F19310, and 20H03552; Uehara Memorial Foundation to M.S., JSPS KAKENHI Grants 16K00228, 18KK0308 and 20H03552; Shimadzu Science Foundation; G-7 Scholarship Foundation; Uehara Memorial Foundation to T.T. We also thank M. Sakurai for secretarial support.

REFERENCES

[1] Joshua H. Jennings, Christina K. Kim, James H. Marshel, Misha Raffiee, Li Ye, Sean Quirin, Sally Pak, Charu Ramakrishnan, and Karl Deisseroth, "Interacting neural ensembles in orbitofrontal cortex for social and feeding behaviour," *Nature*, vol. 565, no. 7741, pp. 645–649, 2019.

[2] Wei Li, Lei Ma, Guang Yang, and Wenbiao Gan, "REM sleep selectively prunes and maintains new synapses in development and learning," *Nature Neuroscience*, vol. 20, no. 3, pp. 427—437, 2017.

[3] Andrea Giovannucci, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, Farzaneh Najafi, Jeffrey L Gauthier, Pengcheng Zhou, Baljit S Khakh, David W Tank, Dmitri B Chklovskii, and Eftychios A Pnevmatikakis, "CaImAn an open source tool for scalable calcium imaging data analysis," *eLife*, vol. 8, pp. e38173, 2019.

[4] Pengcheng Zhou, Shanna L Resendez, Jose Rodriguez-Romaguera, Jessica C Jimenez, Shay Q Neufeld, Andrea Giovannucci, Johannes Friedrich, Eftychios A Pnevmatikakis, Garret D Stuber, Rene Hen, Mazen A Kheirbek, Bernardo L Sabatini, Robert E Kass, and Liam Paninski, "Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data," *eLife*, vol. 7, pp. e28728, 2018.

[5] Igor Gridchyn, Philipp Schoenenberger, Joseph O'Neill, and Jozsef Csicsvari, "Assembly-specific disruption of hippocampal replay leads to selective memory deficit," *Neuron*, vol. 106, pp. 291–300, 2020.

[6] Davide Ciliberti, Frédéric Michon, and Fabian Kloosterman, "Real-time classification of experience-related ensemble spiking patterns for closed-loop applications," *eLife*, vol. 7, 2018.

[7] Logan Grosenick, James H. Marshel, and Karl Deisseroth, "Closed-loop and activity-guided optogenetic control," *Neuron*, vol. 86, pp. 106–139, 2015.

[8] Khaled Ghandour, Noriaki Ohkawa, Chi Chung Alan Fung, Hirotaka Asai, Yoshito Saitoh, Takashi Takekawa, Reiko Okubo-Suzuki, Shingo Soya, Hirofumi Nishizono, Mina Matsuo, Makoto Osanai, Masaaki Sato, Masamichi Ohkura, Junichi Nakai, Yasunori Hayashi, Takeshi Sakurai, Takashi Kitamura, Tomoki Fukai, and Kaoru Inokuchi, "Orchestrated ensemble activities constitute a hippocampal memory engram," *Nature Communications*, vol. 10, 2019.

[9] Deependra Kumar, Iyo Koyanagi, Alvaro Carrier-Ruiz, Pablo Vergara, Sakthivel Srinivasan, Yuki Sugaya, Masatoshi Kasuya, Tzong-Shiue Yu, Kaspar E. Vogt, Masafumi Muratani, Takaaki Ohnishi, Sima Singh, Catia M. Teixeira, Yoan Chérasse, Toshie Naoi, Szu-Han Wang, Pimpimon Nondhalee, Boran A.H. Osman, Naoko Kaneko, Kazunobu Sawamoto, Steven G. Kernie, Takeshi Sakurai, Thomas J. McHugh, Masanobu Kano, Masashi Yanagisawa, and Masanori Sakaguchi, "Sparse activity of hippocampal adult-born neurons during rem sleep is necessary for memory consolidation," *Neuron*, vol. 107, pp. 552–565, 2020.

[10] Eftychios A. Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, Misha Ahrens, Randy Bruno, Thomas M Jessell, Darcy S Peterka, Rafael Yuste, and Liam Paninski, "Simultaneous denoising, deconvolution, and demixing of calcium imaging data," *Neuron*, vol. 89, pp. 285—299, 2016.

[11] Eftychios A. Pnevmatikakis, "Analysis pipelines for calcium imaging data," *Current Opinion in Neurobiology*, vol. 55, pp. 15–21, 2018.

[12] Andrea Giovannucci, Johannes Friedrich, Matt Kaufman, Anne Churchland, Dmitri Chklovskii, Liam Paninski, and Eftychios A Pnevmatikakis, "OnACID: Online analysis of calcium imaging data in real time," *BioRxiv*, p. 193383, 2017.

[13] Lina M. Tran, Andrew J. Mocle, Adam I. Ramsaran, Alexander D. Jacob, Paul W. Frankland, and Sheena A. Josselyn, "Automated curation of CNMF-E-extracted ROI spatial footprints and calcium traces using open-source autoML tools," *Frontiers in Neural Circuits*, vol. 14, pp. 42, 2020.

[14] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter, "Efficient and robust automated machine learning," *Advances in Neural Information Processing Systems 28*, 2015.

[15] Trang T. Le, Weixuan Fu, and Jason H. Moore, "Scaling tree-based automated machine learning to biomedical big data with a feature set selector," *Bioinformatics*, vol. 36, no. 1, pp. 250–256, 2020.

[16] Daniel A Cantu, Bo Wang, Michael W Gongwer, Cynthia X He, Anubhuti Goel, Anand Suresh, Nazim Kourdougli, Erica D Arroyo, William Zeiger, and Carlos Portera-Cailliau, "EZcalcium: Open-source toolbox for analysis of calcium imaging data," *Frontiers in Neural Circuits*, vol. 14, pp. 25, 2020.

[17] Jinghao Lu, Chunyuan Li, Jonnathan Singh-Alvarado, Zhe Charles Zhou, Flavio Fröhlich, Richard Mooney, and Fan Wang, "MIN1PIPE: A miniscope 1-photon-based calcium imaging signal extraction pipeline," *Cell Reports*, vol. 23, no. 12, pp. 3673–3684, 2018.

[18] F.D.W. Radstake, E.A.L. Raaijmakers, R. Luttge, Svitlana Zinger, and

Jean-Philippe Frimat, "CALIMA: The semi-automated open-source calcium imaging analyzer," *Computer Methods and Programs in Biomedicine*, vol. 179, pp. 104991, 2019.

[19] Takashi Takekawa, Hirotaka Asai, Noriaki Ohkawa, Masanori Nomoto, Reiko Okubo-Suzuki, Khaled Ghandour, Masaaki Sato, Yasunori Hayashi, Kaoru Inokuchi, and Tomoki Fukai, "Automatic sorting system for large calcium imaging data," *BioRxiv*, 2017.

[20] Yaesop Lee, Jing Xie, Eungjoo Lee, Srijesh Sudarsanan, Da-Ting Lin, Rong Chen, and Shuvra S. Bhattacharyya, "Real-time neuron detection and neural signal extraction platform for miniature calcium imaging," *Frontiers in Computational Neuroscience*, vol. 14, pp. 43, 2020.

[21] Eran A. Mukamel, Axel Nimmerjahn, and Mark J. Schnitzer, "Automated analysis of cellular signals from large-scale calcium imaging data," *Neuron*, vol. 63, pp. 747–760, 2009.

[22] Johannes Friedrich, Pengcheng Zhou, and Liam Paninski, "Fast online deconvolution of calcium imaging data," *PLoS Computational Biology*, vol. 13, no. 3, pp. e1005423, 2017.

[23] Johannes Friedrich, Andrea Giovannucci, and Eftychios A Pnevmatikakis, "Online analysis of microendoscopic 1-photon calcium imaging data streams," *PLoS Computational Biology*, vol. 17, no. 1, pp. e1008565, 2021.

[24] Joshua T. Vogelstein, Adam M. Packer, Timothy A. Machado, Tanya Sippy, Baktash Babadi, Rafael Yuste, and Liam Paninski, "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *Journal of Neurophysiology*, vol. 104, pp. 3691—-3704, 2010.

[25] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.