Neuroimage Denoiser for removing noise from transient fluorescent signals in functional imaging.

Stephan Weißbach[1,2], Jonas Milkovits[1,2], Michela Borghi[1], Carolina Amaral[1], Abderazzaq El Khallouqi[1], Susanne Gerber[2], Martin Heine[1]

[1] Institute of Developmental Biology and Neurobiology (iDN), Johannes Gutenberg-University, 55128 Mainz, Germany,
[2] Institute of Human Genetics, University Medical Center, Johannes Gutenberg-University, 55131 Mainz, Germany

Keywords: glutamate imaging, functional imaging, iGluSnFR, denoising, deep learning

Highlights

- Neuroimage Denoiser is a deep learning framework to remove noise from functional microscopic recordings, particularly trained and tested for glutamate imaging
- Neuroimage Denoiser balances the removal of noise while preserving the amplitude of responses
- Neuroimage Denoiser operates without re-training for different sensors (when the localization is similar) and recording frequencies

Motivation

Accurate measurements of neuronal activity through functional imaging are critical in understanding mechanisms of synaptic plasticity and learning concerning changes in the molecular composition of single synapses. Traditional denoising methods, such as Gaussian or Median filters, indiscriminately smooth entire recordings, reducing temporal and spatial resolutions considerably. Existing frameworks are not suited to remove noise from glutamate recordings due to the fast dynamics of the sensor. Therefore, a specialized tool for the challenges imposed by glutamate recordings, i.e. faster dynamics, and synaptic localization, is needed.

Summary

We developed Neuroimage Denoiser, a novel U-Net-based model that effectively removes noise from microscopic recordings of transient local fluorescent signals. The model makes the denoising process independent of the recording frequency and the kinetics of the sensor used. The framework is easy to use for denoising and training

and has minimal hardware requirements, thus, making it accessible for an average laboratory to create a custom version specific to their experimental setup. Neuroimage Denoiser significantly enhances the quality of functional microscopy recordings by effectively removing noise, thereby facilitating a more accurate and reliable analysis of neural activity.

Introduction

Noise is a pervasive phenomenon impacting all experimental systems used in neuroscience. Accurate measurement of neuronal activity is crucial for understanding neuronal function, yet noise in imaging systems can severely distort data[1]. With the development of genetically encoded calcium or neurotransmitter sensors, the possibility reading the activity of neuronal subcellular compartments like synapses became feasible, based on fast time-lapse fluorescent imaging. In camera-based setups, noise affects pixels individually without any spatiotemporal relations, often uneven distributed over the field of view. However, the signal from fluorescent reporters that indicates neuronal activity in synapses or somata exhibits a high spatiotemporal correlation.

Traditional methods, such as Gaussian or Median filters, can be used to remove noise from microscopic recordings. While effective to some extent, these filters indiscriminately smooth the entire image, leading to reduced temporal and spatial resolutions[2]. Additionally, the manual selection of filter parameters can be labor-intensive and subjective, often resulting in suboptimal noise reduction.

There has been a considerable effort to remove noise from calcium imaging data and several frameworks have been proposed[3–7]. Notably, Lecoq et al. (2018) developed the deep-learning approach DeepInterpolation[3], a U-Net architecture framework that effectively reduces noise in various biological recordings, including somatic calcium imaging, fMRI, and MEA recordings. DeepInterpolation leverages the data's inherent spatial and temporal relationships to distinguish between true signals and noise, offering a more sophisticated and automated approach to noise reduction[3].

However, commonly used sensors like somatic GCaMP6f, which measure calcium influx, exhibit slower dynamics compared to diffusion-dominated local release of neurotransmitters. Monitoring the synaptic glutamate release needs considerably higher spatial and temporal resolution[8], accessible with camera-based approaches. This can be challenging for a tool that was particularly trained on data acquired using sensors with vastly slower kinetics, i.e. TITL-GCaMP6f, and TIT2L-GCaMP6f-ICL-tTA2[3,9,10]. The glutamate sensor iGluSnFR3.v857.SGZ[11] fluoresces upon glutamate release in the synaptic cleft and subsequent binding to the sensor, providing a much faster signal directly related to neuronal communication[8]. These synaptic signals originate from the fusion of individual transmitter vesicles within the presynaptic membrane and are accompanied by higher background noise compared to the larger somatic calcium signals.

To address these specific challenges, we developed Neuroimage Denoiser, a novel model that effectively removes independent noise from microscopic recordings. Unlike traditional methods, the training and architecture of Neuroimage Denoiser makes the denoising process independent of the recording frequency and the kinetics of the sensor used. This flexibility sets Neuroimage Denoiser apart from previous methods and makes it a versatile tool for various experimental setups, including fast calcium imaging.

Neuroimage Denoiser has minimal hardware requirements, making it accessible for many laboratories to create a custom version tailored to their specific experimental setup. Neuroimage Denoiser enables researchers to obtain high-quality, noise-reduced data without the need for extensive computational resources or expertise.

Material and Methods

Primary hippocampal mouse cultures

Primary mouse hippocampal cultures were generated from newborn pups[12]. Animals were sacrificed according to the European and local government (Rheinland-Pfalz, Germany) regulations for animal welfare. After dissociation with trypsin (Thermo Fisher Scientific; Cat#: 15090-046), cells were plated on poly-L-lysine-coated 18 mm glass coverslips with a density of 70,000 cells per coverslip. After 1 h incubation in Dulbecco's Modified Eagle Medium (Thermo Fisher, Gibco; Cat#: 41966-029/D6429) supplemented with 10% fetal bovine serum and 0.5% glutamate at 37°C, coverslips were transferred to 12-well plates containing 1mL of growth medium composed of Neurobasal-A (Thermo Fisher, Gibco; Cat#: 12349-015) with 2% GlutaMax (Thermo Fisher, Gibco; Cat#: 35050-038), 2% B27 (Thermo Fisher, Gibco; Cat#: 17504-044), 0.1 M sodium pyruvate (MERK, Sigma; Cat#: S8636). Cultures were maintained in a humidified incubator with an atmosphere of 95 % air and 5 % CO_2 at 37°C.

At DIV3-5 neurons were transfected with glutamate sensor iGluSnFR3.v857.SGZ (Addgene; Cat#: 178330)[11] to perform functional imaging experiments, using the calcium phosphate method. To validate the generalizability of the model, one culture was infected with iGluSnFR.S72A (Addgene; Cat#: 106176)[13]. Briefly, the neuron's growth medium was exchanged with 1 mL of pre-warmed Neurobasal-A medium, and the original medium was stored and kept at 37°C. In the meantime, per 18 mm coverslip, 30 μL of transfection buffer containing in mM: 274 NaCl, 9.5 KCl, 1.4 Na_2HPO_4, 15 Glucose, 15 HEPES, pH 7.14 were added to a solution containing 3 µg of DNA and 2 M $CaCl_2$ and 24 μL of H_2O (q.s. 30 μL). The resulting DNA/$CaPO_4$ mix was incubated for 20 min at RT and then added in a proportion of 60 of mix per well. Incubation on neurons was maintained at 37°C until precipitate formation (40 min). Following DNA/$CaPO_4$ incubation, neurons were washed two times for 5 min with 1X HBS buffer containing in mM: 137 NaCl, 4.75 KCl, 0.7 Na2HPO4, 7.5 Glucose, 7.5 HEPES, pH 6.9 and for one time for 5 min with Neurobasal-A medium. Following the

washing steps, the original growth medium was added, and neurons were stored in the incubator at 37°C until live imaging experiments were performed at DIV16 - 19.

Functional Imaging

All live experiments were conducted on hippocampal cultures 16 - 19 DIV at 37°C within a whole microscope incubator. Neurons were perfused with extracellular solution containing (in mM): 145 NaCl, 2.5 KCl, 10 HEPES, and 10 D-glucose 2 $MgCl_2$, 2 $CaCl_2$ (pH 7.4). Network activity was suppressed by adding 10 µM CNQX (Tocris; Cat#: 1045) and 10 µM D-AP5 (Tocris; Cat#: 0106) to the extracellular solution as indicated for the respective experiment. Experiments were performed with an inverted total internal reflection fluorescence (TIRF) setup composed of the eclipse Ti microscope (Nikon) equipped with a 60 x Apo TIRF oil objective (1.49 NA; Nikon). Fluorescence was excited with a combined laser system (Coherent; MPB Communications Inc.). Images were captured by a scientific CMOS camera (ORCA-Fusion C14440-20UP, Hamamatsu) at frame rates between 100 to 1000 Hz controlled by NIS-Elements Advanced Research acquisition software (Nikon). Electrical field stimulation was carried out using two platinum electrodes placed on the coverslip at a 10 mm distance. Extracellular stimulus sequences (0.9 ms; 50 mA; 1 s interval) were triggered by an isolated pulse generator (A&M Systems, Model 2100) and delivered via a stimulus isolator unit A385 (WPI). Images were acquired by image streaming for 2000 frames at 100 Hz within a defined region of interest (ROI) 800 x 800 pixels (100 Hz acquisition rate) or 250 x 250 pixels (1000 Hz acquisition rate) with a pixel size of 72nm. Triggered electrical field stimulation was used to elicit 20 action-potential-like stimuli.

Synaptic ROI and Threshold

Circular synaptic ROIs were selected manually using Fiji (version 1.54f)[14] and set to a diameter of 9 pixels. The average traces along the z-axis of the synapses were extracted by either Fiji (version 1.54f)[14] or Python (version 3.10.14)[15] with roifile (version v2024.3.29)[16].

On the average traces, the threshold was computed using numpy (version 1.26.4)[17] based on the baseline frames up to the first stimulation, using the formula:

$$Threshold(x) = mean(x) + threshold_{mult} * std(x)$$

Whenever the average trace crossed the computed threshold (set by $threshold_{mult}$), this was considered to be a response.

Preparing Training Data

Training deep learning models is sensitive to imbalances between foreground and background[18,19]. The microscopic recording of synaptic glutamate responses is dominated by background due to the small synapses that sparsely respond when defining a responding synapse as foreground. Therefore, the available data is filtered for active regions and frames to remove the background (Figure 1a). We transformed our recording frames into activity maps to filter efficiently. First, the data is transformed

by a pixel-wise rolling window (along 50 frames) z-normalization (Figure 1b). The rolling window adapts for potential bleaching in the recording.

$$z\ score_{frame} = \frac{x_{frame} - \mu_{window_{start}:window_{end}}}{\sigma_{window_{start}:window_{end}}}$$

An activity map is calculated framewise and represents the maximum z-score within a local region for each spatial position in each frame. Each frame is divided into crops of a predefined size of 32 x 32 pixels to compute the activity map. A crop is assigned with the maximum value of the kernel-averaged region, obtained by sliding a quadratic kernel over the crop (step size 1) and computing the average intensity within that kernel at each position (Figure 1c, right). The kernel size corresponds to the expected minimal region of interest (ROI) size, reflecting the spatial extent of the structures under investigation (6 pixels for synapses in our setup). The activity map effectively captures and highlights regions exhibiting high-intensity fluctuations that are potential indicators of neuronal activity, by assigning the maximum averaged kernel value to each crop. Next, these activity maps are filtered for crops that exceed a minimum z-score of 3. Subsequently, the pixel-wise z-score (without using a rolling window) is computed, and the transformed crop is stored in an h5 file for training, using h5py (version 3.11.0).

Neuroimage Denoiser can be run on an entire directory storing multiple recordings and will automatically prepare the training data without manual intervention. Additionally, a memory-optimized version is implemented that necessitates less RAM at the cost of a longer run-time.

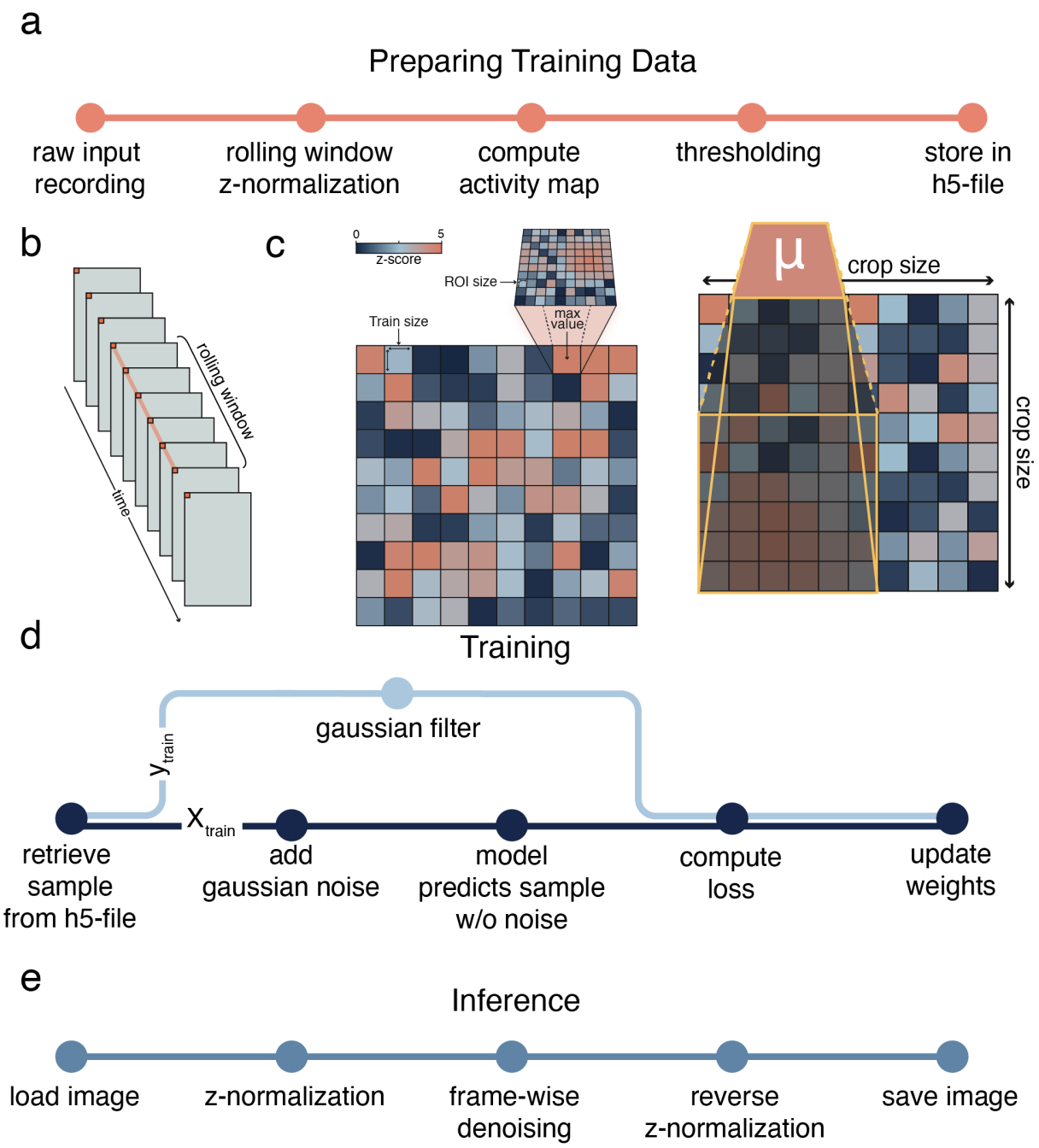

Figure 1 a) Schematic representation of the preparation of the training data. The preparation aims to enrich the foreground (crops of the recording with responding synapses/somata) over the background (non-responding frames or crops) and store them in an h5 file for training. b) Z-normalization for training data preparation is computed per pixel using a rolling window to account for potential bleach occurring during the recording. c) Schematic representation of an activity map that is computed for each frame of a recording during the training data preparation step. The frame is divided into crops of equal size. Per crop, the average z-score per possible ROI of a given size is computed and whenever this averaged z-score exceeds a set threshold, the crop is added to the training data. Left: one frame divided into crops and applied average filter; right: average filter applied of ROI size 6 to a crop d) Schematic representation of the training routine. X_{train} represents the input on which the model makes its prediction, and the result is compared to y_{train}. f) Schematic representation of the inference/denoising process. The inference is done on a z-normalized representation of the recording, which is reversed after denoising.

Model Architecture and Training

The U-Net model[20] is implemented in PyTorch (version 2.1.1.)[21] with a single input and output channel (Figure 1a). The U-Net architecture consists of an encoder and a

decoder. The encoder comprises blocks containing two 2D convolutional layers with 3x3 kernels, ReLU activation, and batch normalization, followed by a 2D max-pooling layer for downsampling. These blocks are referred to as down-blocks. Skip connections connect the corresponding down-blocks to the up-blocks in the decoder without any processing.

The decoder comprises up-blocks, each consisting of a 2D transpose convolutional layer for upsampling, followed by a concatenation with the respective feature map from the corresponding down-block (skip connection). This concatenated output is then processed by two 2D convolutional layers with 3x3 kernels, ReLU activation, and batch normalization. The final layer is a 1x1 2D convolutional layer that maps the feature vector to the denoised image. This architecture allows the network to capture both local and global features, enabling accurate denoising of the input image. Using a fully convolutional model without dense layers allows flexible input image sizes during inference, limited only by GPU memory, overcoming fixed training input constraints.

Training is performed on z-transformed patches containing active synapses for one epoch using a batch size of 64, the ADAM optimizer with a learning rate of 0.0001, on a consumer-grade NVIDIA GeForce 2060 super graphics card with 8GB of vRAM. A grid search for the parameters listed in Table 1 was performed to find a suitable combination of parameters for the training. The model's objective is to reduce noise in the recordings while maintaining the integrity of the peak response amplitudes.

To evaluate the model's performance in preserving the amplitude of neural responses, regions of interest (ROIs) exhibiting clear stimulus-evoked responses were manually identified in the raw recorded data. The peak response amplitudes within these ROIs were measured following the presentation of a stimulus. After training the model, the same ROIs were analyzed in the denoised output, and the corresponding peak response amplitudes were extracted. The model's accuracy in preserving the response amplitudes is quantified by calculating the Mean Absolute Percentage Error (MAPE) using scikit-learn (version 1.5.0)[22] between the raw and denoised peak amplitudes across all ROIs. The correlation between raw and denoised peak amplitudes was calculated using scipy[23].

$$MAPE = 100 * \frac{1}{n}\sum_{t=1}^{n}\left|\frac{peak_t^{raw} - peak_t^{denoised}}{peak_t^{raw}}\right|$$

Additionally, it is crucial that the model efficiently removes noise from the microscopic recordings. This is measured by the standard deviation of non-responding sections of the recordings.

Both metrics (MAPE and standard deviation) are ranked, and a combined score is calculated. The best-performing model is selected based on the minimal combined score achieved (supplementary table 1).

$$score = \frac{(Rank_{std} + 2 * Rank_{MAPE})}{3}$$

Table 1 Model Optimization Parameters: This table presents the various parameters tested during the optimization of the Neuroimage Denoiser model. For each parameter, different variables were tested to determine

the most effective configuration. The "Best performing Model" column lists the values that yielded the best performance based on the evaluation metrics.

Parameter	**Tested Variables**	**Best performing Model**
Loss Function	L1, Smooth-L1, MSE, Huber	Smooth-L1
Use Gaussian Filter on y_{True}	True, False	True
Sigma Gaussian Filter	0.5, 1.0	1.0
Noise Center	0, 0.5, 1.0	0
Noise Scale	0.5, 1.0, 1.5, 2.0, 3.0	2.0

Inference/Denoising

The denoising (or inference) using Neuroimage Denoiser can be done on single images or entire directories containing multiple sub-directories while maintaining the original folder structure. The process is schematically visualized in Figure 1e. First, the input image is loaded in tiff or nd2 format using tifffile (version 2023.12.9)[24] or nd2 (version 0.8.1) respectively and a z-normalization (along the time axis; no rolling window) is applied. The mean and standard deviation per pixel is temporarily saved. Neuroimage Denoiser performs the denoising on the z-normalized frames and afterward, the resulting frames are transformed back into intensity values using the saved mean and standard deviation. The denoised image is saved as a tiff file using tifffile (version 2023.12.9)[24].

Reimplementation of DeepInterpolation

To fairly compare the performance of DeepInterpolation[3] with the approach of Neuroimage Denoiser, we re-implemented the model in PyTorch (version 2.1.1)[21]. Therefore, the model can be trained on crops of active synapses (32 x 32 pixels) and comparability was ensured. The number of frames before and after the target frames was reduced to five frames to account for the faster dynamics of glutamate sensors (Supplementary Figure 1a). The model was trained using the ADAM optimizer with an L1 loss function, and a learning rate of 0.0001 on an NVIDIA GeForce GTX2060 super graphics card.

Data Visualization

Data was visualized using Python (version 3.10.14)[15], matplotlib (version 3.9.0)[25], and seaborn (version 0.13.2)[26]. Schematic drawings were made using Adobe Illustrator.

Results

Denoising glutamate imaging data presents unique challenges compared to calcium imaging, as glutamate sensors are orders of magnitude faster and the synaptic localization results in considerably less foreground signal relative to somatic calcium recordings. Here, we present Neuroimage Denoiser, a U-Net-based imaging denoising

tool (Figure 1a) specifically trained to remove noise from microscopic glutamate time-series recordings.

To curate a balanced training dataset adjusted to the challenges posed by glutamate imaging data, the recordings are automatically screened to identify frames and regions exhibiting active synaptic signals (Figure 1a) in the training preparation step without manual intervention. This screening can be adapted to different ROI sizes representing the size in pixels of the biological structure where the sensor is localized. We set the ROI size to 6 pixels to account for the size of synapses harboring the sensor in our microscopic set-up.

Initially, we explored training a model in the style of DeepInterpolation[3]. DeepInterpolation predicts a target frame based on a set number of frames preceding and succeeding the target frame (n_{pre}, n_{post})[3]. To account for the fast dynamics of the iGluSnFR3 sensor compared to Gcamp6f, we set $n_{pre} = 5$ and $n_{post} = 5$ and the model predicts the target frame in the middle (Supplementary Figure 1a). However, this temporal interpolation approach, while effective for slower calcium dynamics, did not adequately preserve the fast glutamate response and led to artifacts in the amplitudes (Supplementary Figure 1b).

Given the limitations of the DeepInterpolation strategy for capturing these rapid synaptic signals, we turned to the Noise2Noise concept proposed by Lehtinen and colleagues[27]. Their work demonstrated image restoration without ground truth data by training networks to distinguish signals from noise patterns by adding artificial noise to noisy images. Leveraging this approach, Neuroimage Denoiser is trained according to the Noise2Noise[27] method on individual frames and subsequently circumvents the dependence on the recording frequency or the dynamics of the sensor. Therefore, a trained model can be applied to a broad range of sensors that have similar localization independently of the recording frequency.

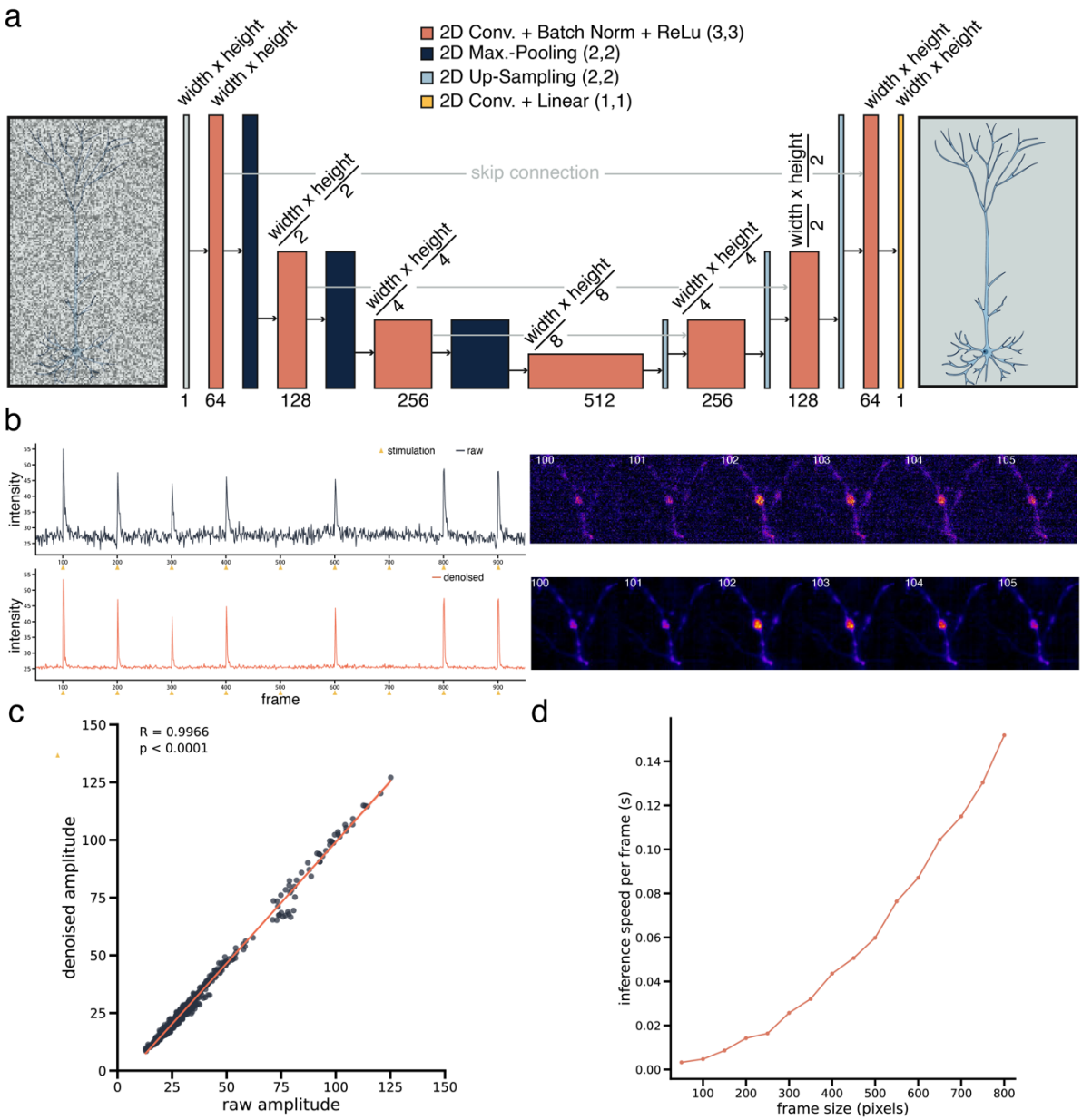

Figure 2 a) Schematic representation of the architecture used for Neuroimage Denoiser is a U-Net[20] with one input and output channel. The data runs through the encoder and decoder part of the network while skip connections preserve untransformed data. Convolutional layers (orange) use a 3 by 3 filter with ReLu activation. The output layer is a linear layer that does not use an activation function. b) Representative area of a glutamate recording acquired at 100Hz. The stimulus is given in frame 100 and the peak response is in frame 102. After denoising, the amplitude height and time points are well preserved, and the noise level is minimal. c) Neuroimage Denoiser denoised amplitude heights of manually selected ROIs tested against the raw amplitude heights, showing a high correlation (R=0.9966, p<0.0001). d) Denoising inference speed (frames per second) tested for frame sizes 50 - 800 pixels (averaged values of three repetitions of recordings with 1,000 frames). The image was processed according to Figure 1e)

We conducted a grid search to identify the optimal combination of training parameters (Table 1) and selected the model that best balanced noise removal and amplitude preservation (Supplementary Table 1). The best-performing model was trained using a smooth L1-loss. The training data included Gaussian noise added with a noise center of 0.0 and a noise scale of 2.0. Notably, higher noise scale and center values yielded models that removed noise more effectively but at the cost of less reliable amplitude

preservation (Supplementary Table 1). The selected model effectively removes noise from the raw recording, both visually as measured in the z-profile of the synaptic ROI (Figure 2b), while preserving the amplitudes well (Figure 2c), as evidenced by a significant correlation between raw and denoised amplitudes ($R=0.9966$, $p < 0.0001$). Neuroimage Denoiser demonstrates fast inference, as tested on three recordings of 1000 frames with varying spatial dimensions (50 to 800 pixels for x and y). The entire process, including normalization and reverse normalization, as shown in Figure 1e, requires between 0.0003 and 0.1519 seconds per frame (Figure 2d). Therefore, denoising 800 x 800 pixels recording of the length of 1,000 frames takes ~152 seconds.

Neuroimage Denoiser enables precise Signal Localization across diverse Recording Conditions

Only infrequent spontaneous activity can be observed when recording a neuronal culture in the absence of electric stimulation. Specifically, it is challenging to differentiate the components at complex synapses, which comprise an accumulation of pre-synaptic inputs and post-synaptic receivers. These synapses often involve multiple release sites in close proximity, and the released glutamate can diffuse from its initial release site, leading to highly correlated signals at neighboring post-synaptic sites (Figure 3a, left). However, the maximum localization of a spontaneous release event indicates the release site. Due to the spatial proximity within these complex synapses, identifying and differentiating post-synaptic receiving sites is difficult when the signal is superimposed by noise (Figure 3a, top). After removing the noise using Neuroimage Denoiser, each of the three distinct sites within the complex synapse becomes visible and can be accurately localized (Figure 3a, bottom).

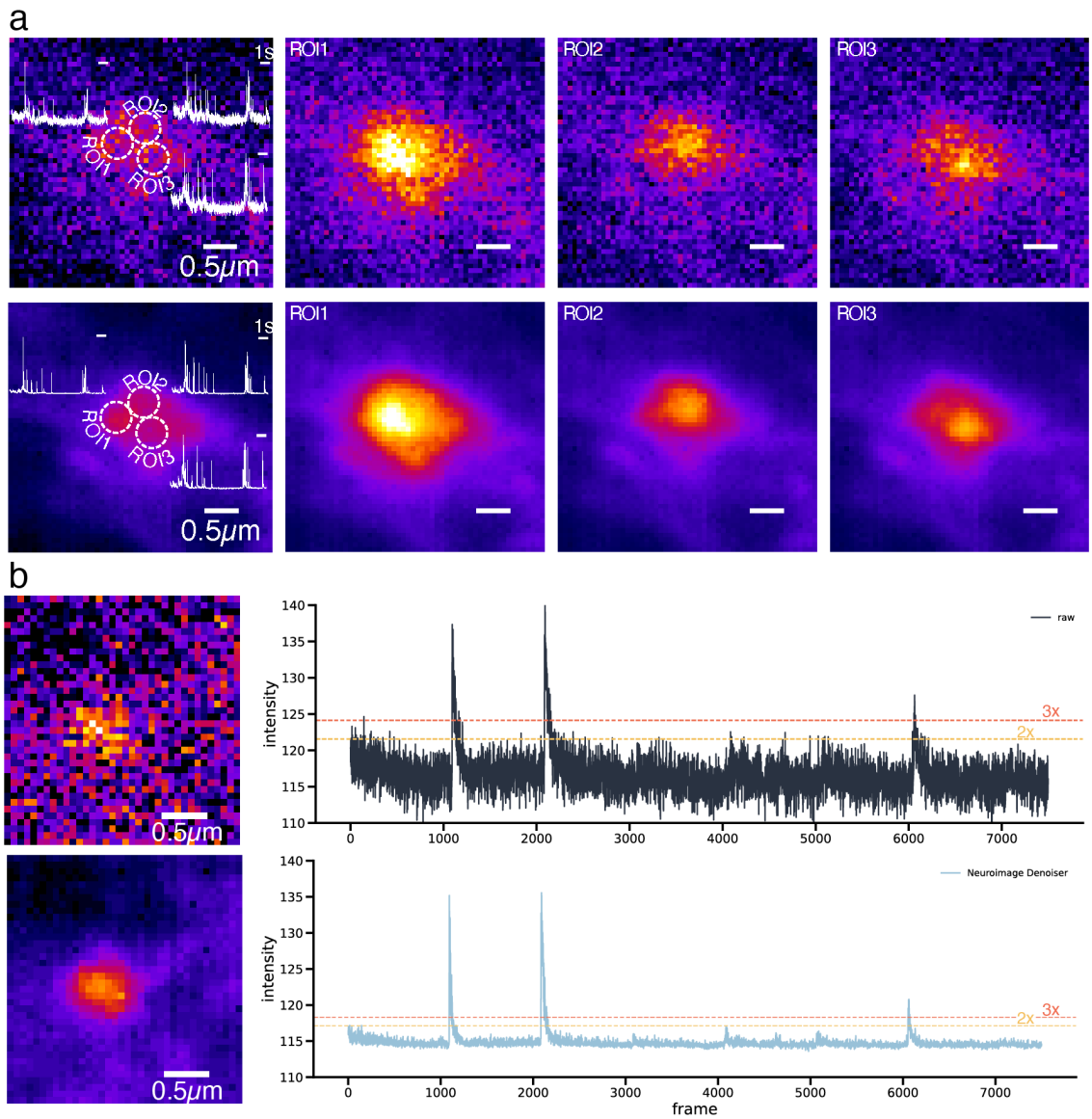

Figure 3 a) Complex synapse with three sites that receive spontaneous input from different pre-synaptic sites. The top row shows the raw recording as it was acquired at the microscope and the bottom row shows the same frames denoised by Neuroimage Denoiser. The three different sites showing spontaneous activity are visible. The traces at these sites are highly correlated due to their proximity. b) Neuroimage Denoiser works on recordings acquired with very high recording frequencies (tested here with 1000Hz) although being trained on 100 Hz recordings exclusively.

While Neuroimage Denoiser was trained exclusively on recordings done with 100 Hz, the model is capable of denoising recordings acquired at significantly higher acquisition rates, such as 1000 Hz (Figure 3b). The ability to acquire data at such high frequencies is valuable for capturing rapid synaptic events and amplitudes with high temporal precision. However, high noise levels often complicate these high-frequency recordings, making it challenging to identify true responses amidst false-positive threshold crossings (Figure 3b, top). The denoised recording, on the other hand, has a considerably improved signal-to-noise ratio (Figure 3b, bottom), like the improvement observed in previous recordings at 100 Hz. Therefore, Neuroimage Denoiser can facilitate the analysis of recordings acquired at very high frequencies without the need to re-train a new model. Acquiring at these high rates opens the door to various

applications, such as analyzing multi-vesicular or asynchronous releases from a single synapse, which would yield variable amplitudes and subsequent peaks, or resolving the precise time from stimulus to glutamate release with higher temporal resolution. Additionally, it enables the study of rapid synaptic plasticity mechanisms and the precise kinetics of neurotransmitter release and clearance. The camera-based imaging will allow the comparison of direct multiple synapses in the field of view, which will outcompete scanning microscopy-based approaches, where only single synapses can be imaged at necessary high acquisition rates (>100 Hz).

Application of Neuroimage Denoiser on alternative Glutamate Sensor to study Synaptic Plasticity

Since Neuroimage Denoiser is not bound to a certain sensor and its associated kinetics but only to its localization, we applied the denoising to recordings of fast iGluSnFR.S72A[13]. This version of the sensor has faster kinetics, but a lower signal-to-noise ratio compared to iGluSnFR3.v857.SGZ[11] particularly due to the high noise level present in the baseline of the recording (until stimulation at frame 100). This noise in the baseline elevates the threshold above which a signal is considered a release event. iGluSnFR.S72A[13] is useful for plasticity studies since the kinetic properties allow the recording of synaptic release events up to 20 Hz pulse interval. Conventionally such recordings are done via patch-clamp[13], which will not allow reaching the single synapse resolution as easily as imaging approaches. Conversely, iGluSnFRs timing and localization of excitatory synaptic inputs[8] enable a more complete picture of the synaptic plasticity of a neuron under a given condition, which is of particular interest for investigating the molecular mechanisms underlying synaptic plasticity. Key plasticity metrics are the paired-pulse ratio, quantifying facilitation or depression of consecutive responses, and the failure rate, measuring the proportion of stimuli failing to evoke a response hence indicating release probability. Nevertheless, due to the low signal-to-noise level and the fast bleaching of the sensor, the recordings are superimposed by noise (Figure 4a, left) that considerably complicates the identification of responding synapses and to differentiate between response and failure within the average trace of a selected synapse (Figure 4b, top). The given example in Fig. 4 illustrates the impact of noise in fast pulse trains. Even though a very small threshold multiplier of two is applied to the raw average trace the responses to the stimulus given at frames 130 and 140 are not detected. Moreover, the noise in the raw data (Figure 4b, top) obscures these responses, while the denoised trace (Figure 4b, bottom) reveals clear and distinct responses upon each of the five stimuli. Thus, the failure rate can be determined to be 0% after applying denoising with Neuroimage Denoiser (threshold multiplier 2), contrary to 40% in the raw trace (threshold multiplier 2). Additionally, when computed with a multiplier of two (yellow dashed line, Figure 4b), the low threshold leads to the detection of false positive responses in the raw trace.

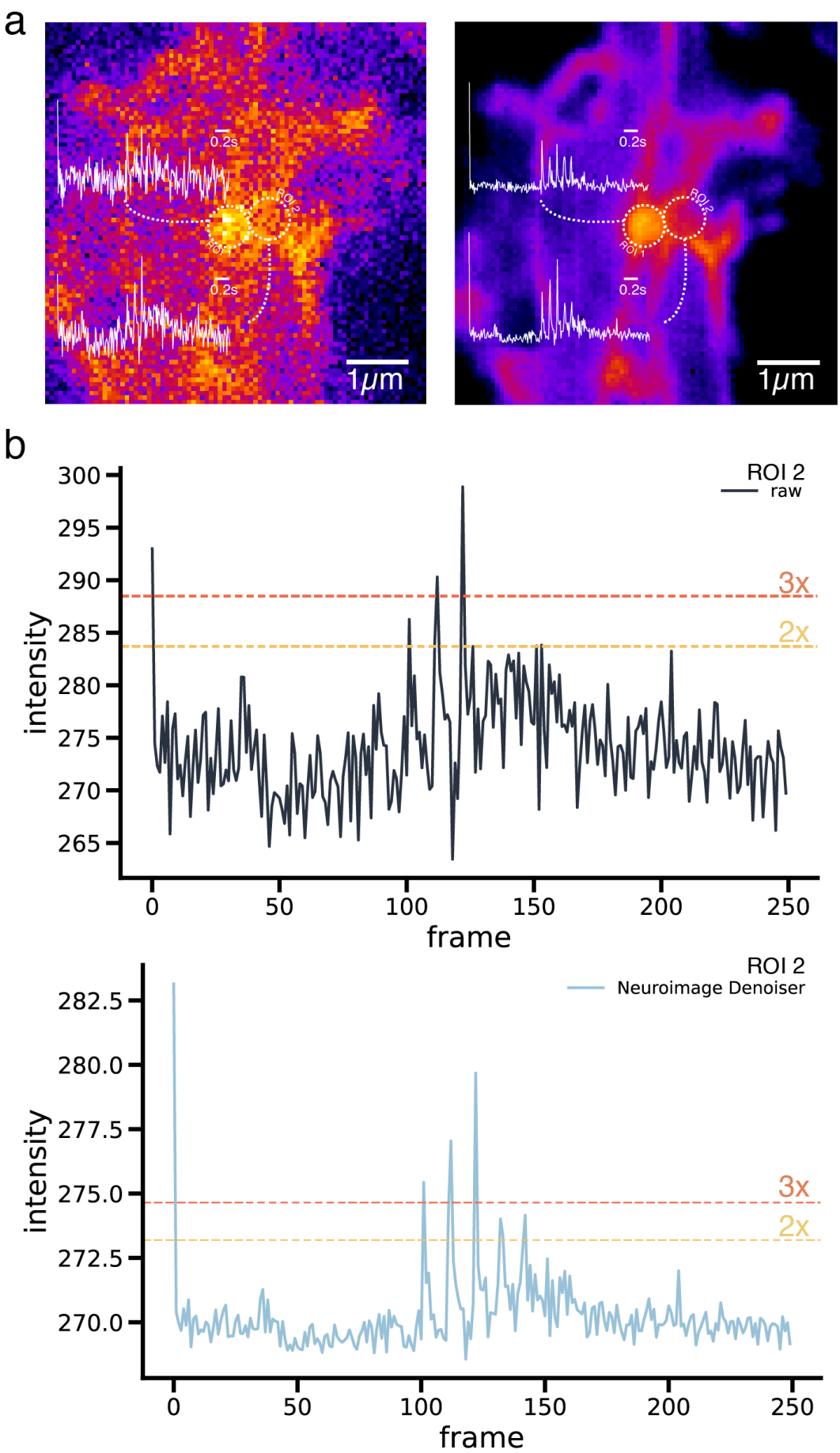
a
0.2s
ROI 2
ROI 1
0.2s
1μm
0.2s
ROI 2
ROI 1
0.2s
1μm
b
ROI 2
raw
3x
2x
intensity
300
295
290
285
280
275
270
265
0
50
100
150
200
250
frame
ROI 2
Neuroimage Denoiser
3x
2x
intensity
282.5
280.0
277.5
275.0
272.5
270.0
0
50
100
150
200
250
frame

Figure 4 a) A crop of the recording with iGluSnFR.S72A. The left panel represents the raw data with high noise levels, while the right panel shows the denoised data using Neuroimage Denoiser. b) Average intensity trace for ROI 2 over the recorded 250 frames. Stimulation is applied at frames 100, 110, 120, 130, and 140. The top graph is extracted from the raw recording and is characterized by significant noise and fluctuations. The bottom graph shows the denoised average intensity trace from ROI 2, where the noise level is substantially reduced, allowing clearer identification of responses to stimuli. Particularly, responses to the stimulation in frames 130 and 140, which are within the noise level in the raw trace, become visible in the denoised trace. The threshold for two low threshold multipliers is indicated as yellow and orange dashed lines, for multipliers two and three respectively.

Discussion

In this manuscript, we presented Neuroimage Denoiser – a U-Net-based[20] deep-learning framework to reduce noise levels in microscopic imaging recordings. By incorporating the concepts of Noise2Noise[27], the Neuroimage Denoiser framework is highly flexible for a wide array of experimental settings and does not require re-training for changed parameters. In the past, researchers developed several sensors to study functional neurobiology for the dynamics of calcium[28], GABA[29], dopamine[30], acetylcholine[31], and glutamate[11,29] on a single synapse level. These recordings are superimposed with noise regardless of the sensor and acquisition rate. Consequently, a unified model capable of denoising recordings when having the same localization would considerably ease the downstream processing. Therefore, we tested our model by applying the denoising routine to changed parameters, i.e. recording frequency (Figure 3) and sensor (Figure 4), and obtained convincing results. The proposed framework is capable of considerably reducing the noise in experiments with fast dynamics, as typical for glutamate imaging, which did lead to artifacts in responses when using previously described deep learning frameworks[3] (Supplementary Figure 1). One of the advantages of Neuroimage Denoiser is its ability to preserve the amplitude of synaptic responses, as evidenced by a high correlation ($R = 0.9966$, $p < 0.0001$) between denoised and raw responses (Figure 2c). Further, the improved signal-to-noise level helps to resolve complex synapses spatially (Figure 3) which can ease the analysis of spontaneous activity and be used as a functional readout for molecular manipulation to define release mechanisms in specific synapses.

We anticipate that Neuroimage Denoiser can be adopted in many laboratories as it has minimal hardware requirements, and the preparation of the training data is automatically balanced for optimal results. The framework is easy to use for data preparation, training, grid search, and inference (denoising) and does not require additional programming knowledge, making it accessible to a broad range of researchers in the field of neuroscience. Although glutamate imaging as a neural recording method can provide rich and diverse datasets, these data have a high complexity with many disruptive elements, such as noise, bright sensor depositions, or vesicular transport of sensor, that make the creation of ground truth data for training a model on responding regions of a recording challenging. Our training data preparation routine offers a way to balance the training data without time-consuming manual intervention and curation. We tested a several potential training parameters (Table 1)

to select the most performant parameter combination, however, users can freely choose a combination suited for their experimental set-up and analysis.
Neuroimage Denoiser is anticipated as a framework for fast responses and is specifically trained to denoise glutamate imaging recordings. However, its flexibility ensures that it is well-suited to work for a variety of typical functional experiments to improve the signal-to-noise ratio.

Code availability

The code is available at https://github.com/s-weissbach/neuroimage_denoiser.

Acknowledgment

The authors acknowledge financial support from the Emergent Algorithmic Intelligence Centre funded by the Carl-Zeiss Foundation. We thank Anita Heine, Nathalie Philipp, and Corinna Werkmann for technical support and excellent hippocampal culture preparation. The authors thank Stanislav Sys for critical feedback and discussion.

Authors contribution

SW and MH conceived the idea and conceptualized the work. MB, CA, AK, and MH conducted experiments. SW conceptualized and implemented the Neuroimage Denoiser framework. SW and JM implemented the data preparation routine. SW created visualizations. SW, MB, and MH wrote the manuscript. SG revised the manuscript. MH and SG supervised the project and acquired funding.

References

1. Wilt, B. A., Fitzgerald, J. E. & Schnitzer, M. J. Photon Shot Noise Limits on Optical Detection of Neuronal Spikes and Estimation of Spike Timing. *Biophys. J.* **104**, 51–62 (2013).
2. Buades, A., Coll, B. & Morel, J.-M. A non-local algorithm for image denoising. in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* vol. 2 60–65 vol. 2 (2005).
3. Lecoq, J. *et al.* Removing independent noise in systems neuroscience data using DeepInterpolation. *Nat. Methods* **18**, 1401–1408 (2021).
4. Pnevmatikakis, E. A. *et al.* Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron* **89**, 285–299 (2016).
5. Malik, W. Q., Schummers, J., Sur, M. & Brown, E. N. Denoising Two-Photon Calcium Imaging Data. *PLOS ONE* **6**, e20490 (2011).
6. Li, X. *et al.* Real-time denoising enables high-sensitivity fluorescence time-lapse imaging beyond the shot-noise limit. *Nat. Biotechnol.* **41**, 282–292 (2023).
7. Chaudhary, S., Moon, S. & Lu, H. Fast, efficient, and accurate neuro-imaging denoising via supervised deep learning. *Nat. Commun.* **13**, 5165 (2022).

8. Marvin, J. S. *et al.* An optimized fluorescent probe for visualizing glutamate neurotransmission. *Nat. Methods* **10**, 162–170 (2013).
9. Madisen, L. *et al.* Transgenic Mice for Intersectional Targeting of Neural Sensors and Effectors with High Specificity and Performance. *Neuron* **85**, 942–958 (2015).
10. Daigle, T. L. *et al.* A Suite of Transgenic Driver and Reporter Mouse Lines with Enhanced Brain-Cell-Type Targeting and Functionality. *Cell* **174**, 465-480.e22 (2018).
11. Aggarwal, A. *et al.* Glutamate indicators with improved activation kinetics and localization for imaging synaptic transmission. *Nat. Methods* **20**, 925–934 (2023).
12. Kaech, S. & Banker, G. Culturing hippocampal neurons. *Nat. Protoc.* **1**, 2406–2415 (2006).
13. Marvin, J. S. *et al.* Stability, affinity, and chromatic variants of the glutamate sensor iGluSnFR. *Nat. Methods* **15**, 936–939 (2018).
14. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
15. The Python Language Reference — Python 3.10.12 documentation. https://docs.python.org/3.10/reference/.
16. Gohlke, C. cgohlke/roifile: v2024.5.24. Zenodo https://doi.org/10.5281/zenodo.11266632 (2024).
17. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
18. Saini, M. & Susan, S. Tackling class imbalance in computer vision: a contemporary review. *Artif. Intell. Rev.* **56**, 1279–1335 (2023).
19. Chen, J., Wu, Q., Liu, D. & Xu, T. Foreground-Background Imbalance Problem in Deep Object Detectors: A Review. in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* 285–290 (2020). doi:10.1109/MIPR49039.2020.00066.
20. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Preprint at https://doi.org/10.48550/arXiv.1505.04597 (2015).
21. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. Preprint at https://doi.org/10.48550/arXiv.1912.01703 (2019).
22. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Mach. Learn. PYTHON*.
23. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
24. Gohlke, C. cgohlke/tifffile: v2024.5.22. Zenodo https://doi.org/10.5281/zenodo.11256883 (2024).
25. Ari, Niyazi & Ustazhanov, Makhamadsulton. Matplotlib: A 2D Graphics Environment. *IEEE J. Mag.* (2014).
26. Waskom, M. L. seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).
27. Lehtinen, J. *et al.* Noise2Noise: Learning Image Restoration without Clean Data. Preprint at http://arxiv.org/abs/1803.04189 (2018).
28. Chen, T.-W. *et al.* Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* **499**, 295–300 (2013).
29. Marvin, J. S. *et al.* A genetically encoded fluorescent sensor for in vivo imaging of GABA. *Nat. Methods* **16**, 763–770 (2019).

30. Sun, F. *et al.* A Genetically Encoded Fluorescent Sensor Enables Rapid and Specific Detection of Dopamine in Flies, Fish, and Mice. *Cell* **174**, 481-496.e19 (2018).
31. Borden, P. M. *et al.* A fast genetically encoded fluorescent sensor for faithful in vivo acetylcholine detection in mice, fish, worms and flies. 2020.02.07.939504 Preprint at https://doi.org/10.1101/2020.02.07.939504 (2020).