applied sciences

Review

Computational Methods for Neuron Segmentation in Two-Photon Calcium Imaging Data: A Survey

Waseem Abbas *,† and David Masip †

Faculty of Computer Science, Multimedia and Telecommunication, Universitat Oberta de Catalunya, Rambla del Poblenou 156, 08018 Barcelona, Spain; dmasipr@uoc.edu

* Correspondence: abbas@uoc.edu

† These authors contributed equally to this work.

Abstract: Calcium imaging has rapidly become a methodology of choice for real-time in vivo neuron analysis. Its application to large sets of data requires automated tools to annotate and segment cells, allowing scalable image segmentation under reproducible criteria. In this paper, we review and summarize the most recent methods for computational segmentation of calcium imaging. The contributions of the paper are three-fold: we provide an overview of the main algorithms taxonomized in three categories (signal processing, matrix factorization and machine learning-based approaches), we highlight the main advantages and disadvantages of each category and we provide a summary of the performance of the methods that have been tested on public benchmarks (with links to the public code when available).

Keywords: datasets; neural networks; calcium imaging; neural activity; matrix factorization; component analysis; deep learning

Citation: Abbas, W.; Masip, D. Computational Methods for Neuron Segmentation in Two-Photon Calcium Imaging Data: A Survey. *Appl. Sci.* **2022**, *12*, 6876. https://doi.org/10.3390/app12146876

Academic Editor: Dusica Marijan

Received: 11 June 2022
Accepted: 3 July 2022
Published: 7 July 2022

1. Introduction

Computational neuroscience has made impressive progress in the modelling of the human brain. Most of the relevant findings have been made studying animal brains [1], which share common properties with the human brain. For decades, neuroscientists used many brain imaging methods for this purpose. Some of the most commonly used are electroencephalography (EEG) [2], positron emission tomography (PET) [3], magnetic resonance imaging (MRI) [4] and functional magnetic resonance imaging (fMRI) [5]. These methods have specific characteristics that make them suitable for certain applications, but they lack two relevant properties: real-time monitoring of neural circuits and single cell resolution. In 2003, Stosiek et al. [6] proposed a protocol based on two-photon in vivo calcium imaging that achieved individual cell resolution and real-time analysis of brain tissues in mice.

Two-photon calcium imaging opened a multitude of research paths in neuroscience, such as in the study of visual cortex [7,8], the olfactory bulb [9] or the cerebellum [10]. Nevertheless, the large scale acquisition of images to be segmented at an individual cell level poses the need for automated neuron segmentation. Two-photon calcium imaging produces 3D volumes where each image plane records a specific located brain region (typically 10–20 µm) at a given time. Neuron spikes are produced at different temporal moments, resulting in large stacks of data where neurons are identified. In addition, neurons may appear overlapped on a single frame, and usually the intensity of the spikes varies significantly, a large portion of the neurons being invisible in most of the time frames. Precise annotation requires taking into account the full stack of spatio-temporal data.

Manual annotation of these data requires a cumbersome effort from expert neuroscientists. The process of manually drawing binary masks on each temporal frame is not scalable, and usually two independent annotators can use different criteria on the same task. A logical step is to let computers learn this task from previously annotated samples, producing reproducible results across research labs.

The pipelines for neural data analysis are generally comprised of three steps apart from the preprocessing. The first step focuses on identifying and segmenting all the neurons visible in the field of view. The second step is aimed at extracting temporal traces of the identified neurons and the third step is aimed at spike deconvolution i.e., inferring the time and frequency of neural spikes. This pipeline is illustrated in Figure 1.

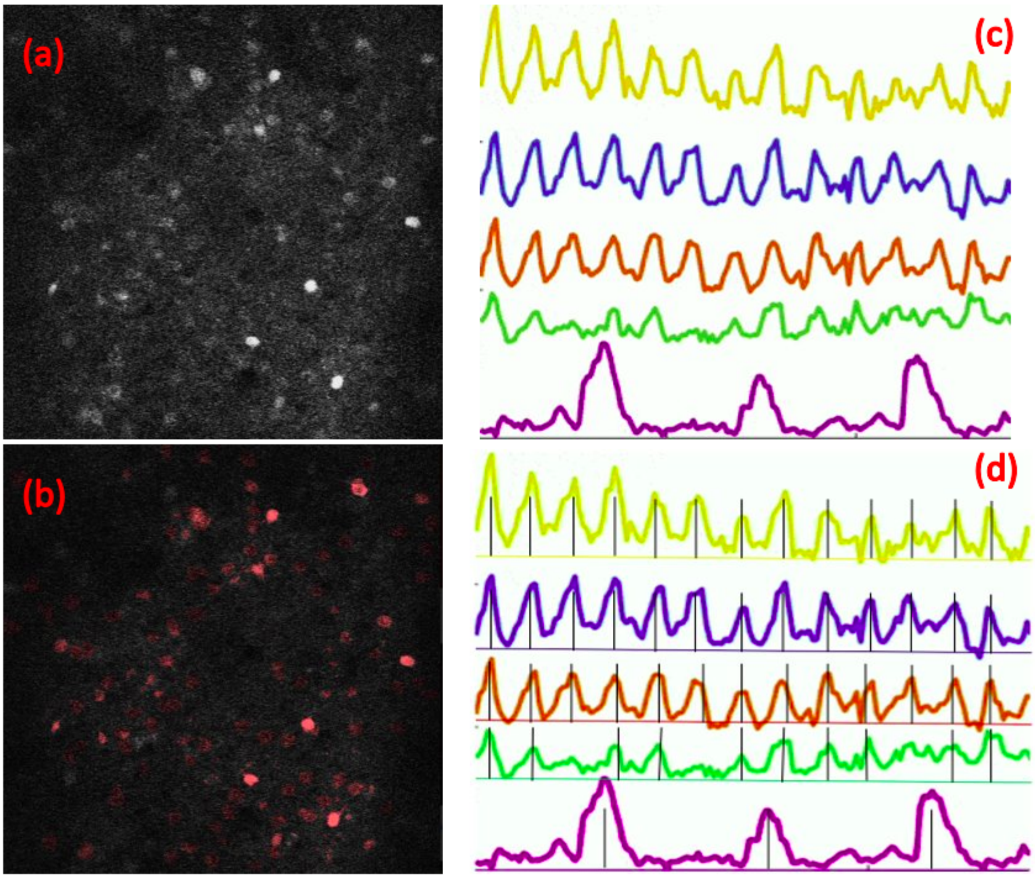

Figure 1. General pipeline followed for calcium imaging data analysis. (**a**) A summary image (see Appendix A.3 for details) representation of a calcium imaging stack. (**b**) Neuron detection/identification/segmentation step where the identified neurons are highlighted. (**c**) Neural traces extraction step, some sample traces are shown. (**d**) Spike deconvolution, inferred spikes are shown for the traces extracted in (**c**).

The number of publications dealing with calcium imaging segmentation has increased significantly. The main motivation of this survey is to provide a comprehensive overview of the recent state-of-the-art works and a proper classification of these methods for its use in neuroscience research. We provide a brief summary of the main features of each method, and a succinct description of their segmentation performance on public benchmarks. This evidence may be useful for neuroscientists that seek out-of-the-box solutions for their particular problems.

This survey reviews some of the most relevant papers on automated neuron segmentation in two-photon calcium imaging (first step of the pipeline illustrated in Figure 1). First, we introduce a taxonomy of the main methods analysed and describe the neural segmentation pipeline. Then we highlight the specific contributions that had an impact in the field. The two main challenges to boost the segmentation are the availability of open code for comparing and improving the current methods and the publication of benchmarks with manually annotated ground truth that allows a fair comparison among algorithms. Section 7 summarizes the results in the context of public benchmarks available. We conclude the paper with a critical discussion of future research.

2. Major Trends

Current neuron segmentation approaches in calcium imaging face the following open challenges: (i) some brain sections present regions with densely located neurons, which can synchronously spike, making the detection and segmentation difficult, (ii) similarly, the summary of the 3D structure of the brain may introduce overlapping neurons, which can confuse the individual cell annotation and (iii) there is a need for automated analysis methods due to the large availability of calcium imaging data, but there is a lack of public large annotated datasets for learning complex models (such as deep learning approaches) that could be used to segment the data available. The survey analyses three main trends followed for neuron segmentation/neural activity detection for calcium imaging data in the literature: neuron segmentation using signal/image processing methods, matrix factorization techniques and machine learning algorithms. The taxonomy of these trends is shown in Figure 2.

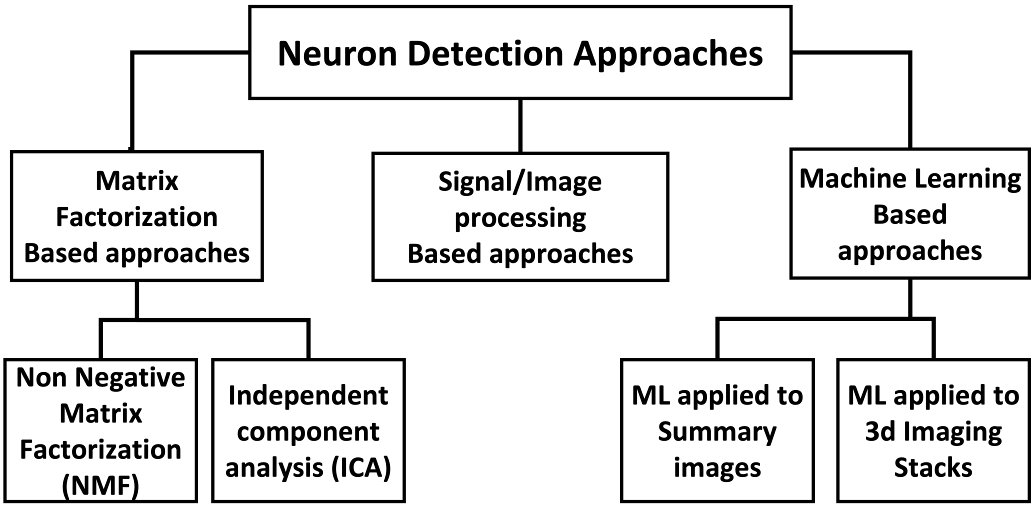
Neuron Detection Approaches
Matrix Factorization Based approaches
Signal/Image processing Based approaches
Machine Learning Based approaches
Non Negative Matrix Factorization (NMF)
Independent component analysis (ICA)
ML applied to Summary images
ML applied to 3d Imaging Stacks

Figure 2. General taxonomy of the major trends and approaches covered in this survey.

Image processing methods use unsupervised signal processing (see Section 4) to filter the neurons on each temporal frame. These methods benefit from the structural features of the neurons' boundaries to highlight them either using convolutions with specific filters, mathematical morphology filters, graph based-clustering, pixel-value thresholding or a combination of them. Most of the approaches which fall in this category rely on some form of background estimation. Since neural activity is recognized by looking for compact spatial regions which exhibit different calcium indicator dynamics than the surroundings, background estimation is a relatively efficient approach to filter out these regions [11–13]. The underlying assumption is that calcium indicators will always emit a baseline fluorescence, even when the neurons are not active. An effective background estimation method which caters to this baseline fluorescence and random noise can therefore quickly isolate any active neurons.

Another particular case of unsupervised learning is the use of matrix factorization methods [14–16]. These methods are based on the assumption that the image is generated by a linear combination of a set of latent variables, which usually correlate with the neuronal signal. Matrix factorization has been widely used in many computer vision tasks, such as face classification [17], unsupervised image denoising [18] and hyperspectral image classification [19]. This can be further sub-divided into two main categories, independent component analysis (ICA) (see Appendix A.2 for details) and non-negative matrix factorization (NMF) (see Appendix A.1 for details). In ICA, the spatial and temporal footprints of the neurons are estimated by solving the matrix factorization framework for two independent matrices, one for spatial footprint and the other for temporal traces. In NMF, the matrix factorization framework is solved in a constrained manner by borrowing one key fact from the physics behind neural florescence i.e., the factorized matrices for spatial footprint and temporal traces cannot contain negative values since a negative value in either spatial footprint or temporal traces is not physically possible. Section 5 reviews some of the most relevant approaches.

The third ensemble of approaches relies on supervised methods. Supervised methods use either manually or anatomically annotated masks to train a robust classifier that performs the neuron segmentation task. These methods usually require large amounts of labelled data, and obtain improved generalization accuracies. With the advent of deep learning methods, two main architectures emerged as the most prominent: convolutional neural networks (CNNs) for pixel classification [20] and U-Net architectures for dense segmentation [21]. These two methods are then applied to the spatio-temporal calcium imaging stacks in two approaches. In the first approach, the temporal dimension of the spatio-temporal stacks are collapsed by calculating summary images (see Section 4) and then the supervised methods are applied to the summary images. This approach has the advantage of speed but suffers from the loss of information since summary images might not adequately represent two types of neurons: those neurons which fire very slowly and those neurons whose emitted fluorescence is not strong enough when compared to the baseline fluorescence/background. Moreover, this approach also struggles with those neurons who are separated in the z-plane but appear overlapped in the summary images since the z-plane (temporal dimension) is collapsed. The supervised methods are directly applied to the 3D spatio-temporal stacks. This approach can detect overlapping neurons but they are heavily reliant on appropriate masks and the length of the spatio-temporal stacks. Moreover, these methods also tend to be computationally more expensive and more difficult to train. Section 6 reviews the most recent methods.

It is to be noted that, on a functional level, signal/image processing-based methods and matrix decomposition-based methods are more similar to each other and more different than machine learning-based approaches as machine learning-based approaches are mostly supervised and many of them need careful annotation of the data. Moreover, these three main categories can be further sub-divided into sub-categories based on the specific approach being taken. For instance, signal/image processing-based approaches can be further sub-divided into two sub-categories: one which treats the temporal traces of each pixel as independent signals and another which treats calcium imaging slices/stacks as 2D structural elements. Matrix decomposition-based approaches can also be sub-divided into matrix factorization-based approaches and component analysis-based approaches. Machine learning-based approaches can be sub-divided into two main functional sub-categories: one which processes calcium imaging stacks one slice at a time and the second which treats calcium imaging stacks either as a spatio-temporal stack or a 3D volume. We decided not to sub-divide these three main approaches into sub-categories to keep a holistic overview and maintain a sense of overall comparison between these three main categories.

Apart from automated segmentation of neurons, tools and pipelines are also developed for manual annotation of neurons or in some cases, fine tuning the automated annotation [22–24]. In some cases, the ROI detection part is manual and calcium spike extraction/detection is automated [24]. We left these approaches out of this survey since they are out of the scope for this survey paper.

3. Neural Activity Segmentation Pipeline

Neural activity/neuron segmentation is a part of the neural data analysis pipeline. The broad neural data analysis pipeline for calcium imaging data includes pre-processing, neuron detection/segmentation, spike inference and spike deconvolution [25,26]. The pipeline does not end at neuron detection/segmentation since the activity of the neurons also needs to be deconvolved by estimating when a neuron fires. In this article, we are dealing only with a part of the neural data analysis pipeline i.e., neuron segmentation. Therefore, we will only explain the pipeline steps necessary for neuron segmentation.

3.1. Motion Correction

The field of view in calcium imaging datasets is not necessarily stationary. Motion artefacts in these datasets arise from natural brain movement and equipment movement. In small areas this motion can be approximated as rigid, and can be corrected using standard template-based registration methods [27]. However, it can also be non-rigid when the field of view is large or brain movement is faster than the raster movement of scanning equipment. Therefore, the first step of neuron detection/segmentation pipelines is to correct any motion artefacts in the datasets. Different approaches have been proposed for motion correction but their in-depth analysis is out of the scope of this article [28–30].

3.2. Filtering

This step involves removing any unwanted noise in the datasets [31]. The noise can come from different sources, such as the neuropil, background fluorescence, equipment or non-uniform staining by calcium indicators. The noise can be reduced by different approaches such as percentile filtering, homomorphic filtering or custom designed pipelines. The detailed analysis of these methods is out of the scope of this paper.

3.3. Source Extraction (Neuron Segmentation)

The next step of the pipeline is source extraction which can either be treated as a neuron segmentation or neuron detection problem. In neuron segmentation approach, the neurons are segmented by identifying every pixel that belongs to a neuron in all the available frames. The segmentation method should ideally produces a mask for each neuron with the neuron as the foreground and everything else as the background. In neuron detection approaches, only the position of each neuron is estimated without identifying every pixel belonging to a neuron. This can also be viewed as neuron location approach and can be used as a pre-processing step for more advanced neuron segmentation approaches.

3.4. Spike Deconvolution

The final step of neural data processing is spike deconvolution, which involves inferring the spike trains of all the segmented neurons. When all the neurons are located, first their activity traces are extracted by averaging the fluorescent values inside the neurons. Since change in the calcium indicators concentration is not an instantaneous impulse, but rather a dynamic process in which the calcium indicators concentration changes quickly at the beginning and then settles down in some time, the actual spike can be represented by a decaying signal in the extracted fluorescence trace. Spike deconvolution aims to extract the spike train from this somewhat continuous fluorescence signal. The detailed analysis of spike deconvolution is out of the scope of this paper [32–37].

4. Signal/Image Processing-Based Methods

In this section, we will cover all those neuron segmentation approaches that are based on signal/image processing. The segmentation problem is approached either from a signal fingerprinting perspective by looking for specific fingerprint of the signal or object localization perspective. We have summarized different signal/image processing-based neurons segmentation methods in this section.

Liu et al. proposed a gradient vector fields-based cell segmentation and localization approach for calcium imaging data acquired from Zebra fish [38]. First, they compute a gradient vector field of the input frame from a calcium imaging dataset. As pixel values of gradient vector fields represent the local gradient flow in the image, the gradient vector fields are not smooth. Therefore, after calculating the fields, they apply a physical model to diffuse the gradient vectors on image boundaries, resulting in a smoother gradient flow field. Given the diffused gradient flow image, for each point, they then determine a set of positive and negative points along the gradient direction. The positive points are found by considering the points in a given radius along the positive gradient direction and same for negative points. These positive and negative points are used to create orientation and magnitude images. For each pixel, the positively voted points in a given radius are incremented by 1 and the negatively voted points are decremented by 1. For magnitude voting image, the positively voted points are incremented by magnitude of gradient multiplied by a monotonically decreasing Gaussian and the negatively voted points are decremented by same amount. After finding magnitude voting image and orientation voting image, non-maximum suppression is applied. Then a distance proportional to the radius of a neuron is used to find local maxima, which represent the nuclei of the circular neurons. To reduce false positives, mean-shift clustering is used by taking the colour and pixel intensities into account. The authors have validated their approach on two datasets. In the first case, they applied their pipeline to count neurons on retinal sections stained with DNA-binding dye. They reported a false positive rate of 4.2% and a false negative rate of 4.5% against a human expert. In the second case, they applied their approach on retinal sections stained with a different dye. They reported a false positive rate of 6.3% and a false negative rate of 5.2% against a human expert. In a hindsight, their approach achieves impressive results, but it is based on the assumption that neurons have a circular shape which might not be true all the time. Moreover, this approach cannot isolate overlapped neurons. Additionally, the approach is not tested on datasets acquired under diverse conditions and does not take the temporal activity of the neurons into account.

Patrick et al. developed SIMA, an open source Python package that facilitates common analysis tasks related to fluorescence imaging [39]. The package can be used for motion correction, ROI (region of interest) segmentation and signal extraction from calcium imaging stacks. The package has a GUI (graphical user interface) for interactive editing of the ROIs as well. The segmentation sub-routines are based on normalized cut approach. The normalized cut approach [40] segments the image by an iterative process. At each stage, a subset of the image pixels is split into two new subsets in such a way as to minimize a penalty that depends on a set of connection weights between the pixels until a termination criterion is met. Two factors determine the resulting cuts: (1) the connection weights between pixels, and (2) the termination criterion. The package developed by the authors implements two versions of the normalized cut. In the first version, the connection weights between two pixels are logarithmically related to the correlation and distance between the pixels. In the second variant of the normalized cut, the connection weights also take into account the maximum intensity pixels along the line connecting the two pixels. The termination criterion for the iterative partitioning of the image depends on the number of pixels in the region and the normalized cut penalty for the next potential partitioning. After the normalized cut procedure, Otsu's method is used to for calculating the threshold of each partition [41]. Then morphological opening followed by closing is used to fill the gaps and regularize the shape of ROIs. Finally, a minimum size and circularity criterion is applied to all the ROIs to exclude any small blobs. The segmentation sub-routines were tested on 37 datasets of two-photon calcium imaging data acquired from pyramidal cells in hippocampal area. The datasets contained 4575 frames in total with spatial dimensions of 128 × 128, acquired at a frame rate of 6.6 Hz. A ground truth was generated by a human expert by visual inspection of all the frames. To determine if a segmented ROI represents a true detection, the Jaccard index was used with all the ROIs with a Jaccard index above 0.25 being considered as true detection. The authors report a false negative rate of 12% ± 2% and a false positive rate of 20% ± 5%. The segmentation sub-routines are computationally efficient to be implemented for real-time segmentation; however, they do not take any temporal context into account. Moreover, they cannot deal with the overlapping neurons in the z-plane and they also cannot deal with the odd doughnuts shaped neurons which appear in datasets acquired by some of genetically encoded calcium indicators.

Mohammed et al. proposed an integrative approach for analysing hundreds of neurons in awake-behaving mice in vivo [42]. This approach includes ROI detection, ROI merging and calcium trace analysis. First, the frames in a calcium imaging stack are standardized using z-scores. An adaptive threshold is then applied on all the pixels in a frame to produce a binary mask. These binary masks then processed using morphological operations to find and label connected components within each frame. This is based on the intuition that neuron pixels are significantly brighter due to activation of GCaMP. The initial threshold was adjusted based on the number of non-zero pixels detected with each threshold. The single-frame ROIs are then merged to create multi-frame ROI by clustering all those single-frame ROIs together based on the proximity of their centroids, as well as the proximity of their bounding boxes. The multi-frame ROIs are then used to extract calcium traces. The approach is tested on calcium imaging data acquired from the CA1 pyramidal cell layer of head-fixed awake mice. The images were acquired at 20 Hz with a spatial resolution of 1024 × 1024. To quantify the ROI detection performance of their algorithms, the author manually inspected each ROI from example dataset. The authors have reported that their ROI detection sub-routine was successful in image regions with less density of ROIs and struggled in image regions with higher density of ROIs. The detection sub-routine cannot handle overlapping neurons in z-plane. As this approach is not tested on any public dataset, we cannot establish its robustness.

Spaen et al. proposed a novel combinatorial approach for identifying neurons in calcium imaging data [43]. The algorithm identifies cells by finding distinct groups of highly similar pixels in correlation space. The authors based their approach on the observation that the pixels in spatial footprint of a neuron have similar temporal signal. They used a graph-based model where nodes represents the pixels and edges represent the connections/similarity relations between pixels. They assigned a value close to 1 if the pixels are similar and close to 0 if the pixels are not similar. Then they iteratively identified one neuron at a time until all the neurons are identified. In the beginning, they selected a superpixel as seed. Then they selected negative seeds by uniformly picking pixels from a circle of sufficiently large radius. All the pixels within the radius are compared with the seed to eliminate non-neurons pixels. Although finding superpixels is efficient and all of them can be used as seeds, this process is computationally not efficient. Therefore, the authors considered only 30–40% of all the superpixels with highest average correlation as seeds. They computed correlation space to find similarity between pixels. To compute the value of a pixel p_i in correlation space, they find correlation of temporal trace of the pixel p_i with the temporal traces of all the pixels surrounding it which are inside a bounding box with the pixel p_i in the centre. Then, they averaged all the correlation values to calculate the value of pixel p_i in the correlation space. The authors have tested their approach on the Neurofinder benchmark. The achieved an average F1 score of 59.6%, an average recall of 62% and an average precision of 65.8% on the test datasets provided in the Neurofinder challenge.

Guan et al. proposed an open source MATLAB toolbox which they call *NeuroSeg*, for automated analysis of two-photon calcium imaging data [44]. The toolbox includes a custom designed cell detection sub-routine based on Gaussian and Laplacian filters. First, they normalized the image by min-max normalization. Then they use a 2D median filters with a kernel size of 5 for noise reduction. They based their neuron detection approach on the observation that the neurons have circular or elliptical blob like shapes. Therefore, they opted for the gLog algorithm proposed by Xu et al. [45]. They apply the gLog algorithm at different scales to detect neurons of different sizes. The size and orientation of the gLog filter kernels are predefined. These kernels were convolved with the calcium imaging data, thus generating response maps for all the kernels. Then, seeds were initiated by searching local maxima in response maps. An adaptive thresholding algorithm was then used to generate a binary mask of the seeds with cell regions represented by a logical 1 and background represented by a logical 0. Those seeds which were in close proximity were merged. To segment neurons, square windows were drawn with the seeds at their centres. The size of the windows was determined by the size of the kernel which generated to response map of the seed. Next, they calculated two weighting matrices. The first weighting matrix is a Gaussian kernel with an elliptical plateau region whose scales and orientation are determined by the scales and orientations of the original gLog kernels. The second weighting matrix is proportional to the Gaussian gradient of the window and normal vectors pointing from the pixels to the seed. These weighting matrices are multiplied with the square window and the result is binarized. The final binarized window contains the neuron which generated the seed point in first place. They tested their approach on an in vivo two-photon calcium imaging dataset obtained from a C57/BL6J male mouse's cortical neurons with varying view fields. They compared the ROIs detected by their approach to manually annotated ROIs. To quantify the level of agreement between manually annotated ROIs and automatically detected ROIs, they used dice similarity coefficient (DSC) and Hausdorff distance (HD) [46]. They reported a DSC of 0.83 ± 0.02 and an HD of 12.91 ± 1.49. They also reported a precision of 87.5% ± 6.6%, a recall of 89.3% ± 5.3% and an F1 score of 88.1% ± 4.0%. Although the approach is simple and analytically tractable, it cannot deal with overlapping neurons. Moreover, it might also find it hard to detect the doughnut-shaped neurons in which calcium activity is concentrated on the borders.

Romano et al. proposed an open-source toolbox for pre-processing, neuron segmentation, trace extraction, estimation of significant single-neuron single-trial signals, mapping event-related neuronal responses, detection of activity-correlated neuronal clusters and exploration of population dynamics in two-photon calcium imaging data [47]. The toolbox includes a neuron segmentation sub-routine which is based on morphological operations. First, a summary image is obtained by averaging the stack across temporal dimension. Then the summary image is spatially normalized by first filtering it by a 10% order statistic filter and then by a 90% order statistic filter. The two resultant images are used for min-max normalization of the summary image. After normalization, the user sets the thresholds for the cell body, the soma and the neuropils which will segment the summary image into binary masks, one for soma and one for neuropil. Both of the masks are then imposed as regional minima to the image complement of the summary image. Finally, watershed transformation is performed to obtain the ROI boundaries. Moreover, the associated GUI lets the user manually curate the ROIs too. The authors have tested their ROI (neurons) segmentation approach on five datasets obtained from different regions of a mouse brain under different conditions (Dataset 2 sampled at 30 Hz with spatial dimensions of 256 × 256, Dataset 3 sampled at 7 Hz with spatial dimensions of 256 × 256, Dataset 4 sampled at 1 Hz with spatial dimensions of 512 × 256, Dataset 5 sampled at 100 Hz with spatial dimensions of 232 × 242, Dataset 1 sampled at 100 Hz). They reported a true detection rate of 85–90%. It is to be noted that the toolbox has not been tested on any public benchmarks, so we cannot establish its robustness. Moreover, the neuron segmentation approach employed in this toolbox cannot separate overlapping neurons.

Tomek et al. developed an open source MATLAB toolbox for pre-processing, neuron segmentation and signals extraction in real time which they call *Search for Neural Cells Accelerated or SeNeCA* [48]. The segmentation algorithm is based on a combination of local thresholding and a watershed method which is interrupted properly. First, the input image is filtered with a blur or disk filter. Then a light averages (LA) matrix is generated by placing average of surrounding pixel intensities at the original pixel spatial position. At this point, the mask is also initiated. The user sets the higher light threshold coefficient and the lower light threshold coefficient while the actual thresholds at each pixel are the product of a particular coefficient and the pixel value of LA matrix at same spatial coordinates. If a pixel value of this product image exceeds the higher threshold, it is considered an intracellular pixel and represents a possible origin for the watershed wave. A flooding wave is started from each point (Meyer's flooding algorithm [49]). The lower light threshold stops the flooding. The pixels are continuously updated in the output mask. The neuron cells are numbered. The authors propose to mark the edge of the nth neuron as n whereas the inside of the nth neuron is marked by $-n$. Finally, cells in the output mask are eliminated if their size is greater or smaller than the expected size. The authors have tested SeNeCA on three kinds of data: synthetic, in vitro and in vivo. To quantify the neuron segmentation performance, the neurons segmented by SeNeCA are compared to the neurons annotated by human experts. Each of the dataset contained 100 images and SeNeCA was tested on a randomly selected image. The authors calculated four kinds of errors to quantify performance: split: when two numbers are assigned to same cell; merged: when two cells are treated as one; spurious: a false positive; missing: a false negative. SeNeCA detected 48.7% false positives, missed 5.9% of the cells and split and merged errors were minimal. In case of synthetic data (dimensions of $256 \times 256 \times 25$), SeNeCA made $1.9\% \pm 0.6\%$ merged error, $7.0\% \pm 2.5\%$ false positives and $4.0\% \pm 3.4\%$ false negatives. SeNeCA is open source, computationally efficient and tractable. However, like most of its counterparts in this category, it fails to take the temporal dimension into account. If the neurons are densely populated, SeNeCA might see a lot of merged errors since it cannot deal with neurons overlapped in the third dimension.

Stephanie et al. proposed a level set-based neuron segmentation approach in two-photon calcium imaging data which they call *ABLE* [50]. The approach treats every neuron and its immediate surrounding as a union of two sub-regions, the neuron as one sub-region and the background as another sub-region. These sub-regions are separated by a boundary and both of them can be represented by their own feature vectors. In this case, the authors consider the average time-courses of the sub-regions as the feature vectors. The optimal partition or the boundary is the one which minimizes the dissimilarities between a pixel time course and the average time-course of the region it belongs to. The optimal boundary is found by iteratively evolving its shape based on the dissimilarity function. The boundary is an active contour. As it gets updated, the interior and exterior of the neuron also get updated. In this article, the authors considered the union of all background sub-regions as one global background sub-regions. They used a level set method to evolve the active contours. First, the temporal dimension of the image stack is collapsed by computing a correlation image. Then local maxima are found in the correlation image which are considered as neurons at the initialization stage so there boundaries become the active contours. The active contours are iteratively updated until either maximum number of iterations is reached or the number of pixels added or removed from the interior is smaller than a threshold. If the neurons are sufficiently closed, they are merged. The authors have tested their approach both on synthetic data and the Neurofinder challenge benchmark. They considered a cell as true positive if its centroid is within five pixels of the centroid of its manually annotated counterpart. To quantify the performance, they used F1 score. On synthetic data, ABLE achieved an F1 score > 99%. On the Neurofinder challenge dataset, ABLE achieved a precision of 67.5%, a recall of 67.5% and an F1 score of 67.5%. ABLE is flexible since no priors are required. Additionally, it only needs local information to evolve a contour. However, like any level set method, the performance of ABLE is dictated by the quality of initialization.

Ellefsen et al. developed an interactive MATLAB/Python toolbox for segmentation, analysis and visualization of calcium imaging data [51]. The toolbox has a custom neuron segmentation sub-routine. The toolbox accepts image stacks as inputs. The input image stack is first filtered to remove flash artefacts. The user is then prompted to select an area of the imaging plane containing no neurons. The average intensity of this area is used to create a background frame. The background frame is subtracted from all the frames in the image stack. Then the resulting image stack is spatially filtered with a Gaussian filter. The spatially filtered image is temporally filtered using a band-pass filter which removes the high frequency photon shot noise and low frequency baseline fluorescence. A user specified boxcar window is used to find the minimum value of preceding frames. This minimum value is then subtracted from the pixel values of the current frame. Those pixel values which are higher than a threshold are set to 1 while the remaining are set to 0. This step generates a 3D Boolean matrix. True pixels close in space and time are grouped together. Afterwards, the Boolean matrix is used to create three dimensional boxes encompassing unique events. The authors have tested their algorithm and routines on a synthetic dataset and an in vitro calcium imaging dataset. The synthetic dataset was generated by creating a neurons firing sequence and adding this to a baseline fluorescence at known locations, at different amplitudes and with different neurons sizes. An image stack of 1200 frames was created using this method. The other dataset was acquired from neuron activity at localized sites in human neuroblastoma SH-SY5Y cells cultured in vitro. The image stack was acquired at 420 frames per second. The ground truth was created by visually inspecting the stack and localizing neural activity. The authors report a maximum false positive rate of 0.007 events per frame (one false positive per cell every 3 s). The authors have not reported the performance of their toolbox on benchmarks, therefore it is hard to evaluate the robustness of their toolbox on diverse datasets. Moreover, the calcium events localization routine in their toolbox seems to ignore the occurrence of overlapping neurons, which can significantly drive the performance down.

Prada et al. developed an open-source bioinformatics tool for automatic extraction, counting and localization of calcium signals in two-photon calcium imaging data [52]. They specifically targeted calcium signals whose amplitude is close to background fluorescence levels. The tool interactively asks the users to enter three parameters: grid window size, signal to noise ratio (SNR) and signal average threshold (SAT). The SNR value defines the threshold separating calcium events from noise and SAT value is used to separate signal from the background. First, the input image stack is divided into multiple grid cells. Then, average calcium fluorescence is calculated for the grid window and its amplitude is compared with the SAT value. Wavelet transform is used to localize the peak in the calcium signal. It is based on the assumption that calcium signal has three components, the peak (calcium transient event), the baseline and the noise. The authors have tested their tool on hippocampal neurons prepared from CDI mice of either sex. The authors reported an average true positive rate of 93% and an average false positive rate of 2.5%. They have not reported any false negative rate. Moreover, the tool has not been tested on a benchmark dataset, so we cannot establish its robustness. Additionally, temporal information is only used once the neurons are detected (grid cell containing neurons); therefore, this tool cannot effectively deal with spatially overlapping neurons.

Simon et al. proposed an adaptive thresholding-based cell segmentation approach which they call "ACSAT (Automatic Cell Segmentation by Adaptive Thresholding)" ([53]). This approach is optimized for cell segmentation in large datasets. First, they collapse the 3D calcium imaging stack onto a 2D image by taking a temporal average. In the second step, ACSAT detects ROIs at various threshold iteratively. In the first iteration, an initial threshold is used to segment potential ROIs from the background. The locations of these ROIs is saved and then the ROIs are eliminated. Next, the threshold is lowered for next iteration. The thresholding is applied both at global level and at local level. The global thresholding segments the ROIs from the background whole the local thresholding separates overlapping ROIs. ACSAT has been validated on a simulated dataset with a recall generally higher than 80%, however, the authors report that the precision and recall fall if the SNR is below 20 dB. ACSAT has also been validated on an actual dataset acquired from the hippocampus of a mouse. It achieved a precision of 71.2% and a recall of 74.9%. On another dataset acquired from the striatum, it achieved a precision of 51.1% and a recall of 75.8%. The performance figures of ACSAT are decent and it is simple enough to be understood and computationally not very expensive. However, it cannot deal with those neurons whose temporal activity is sparse and might not register strongly in the mean image. Moreover, it can also not deal with those overlapping neurons that have similar fluorescence levels.

5. Matrix Factorization-Based Methods

One of the most popular approached for calcium data segmentation is the use of matrix factorization techniques. These techniques aim to solve the neuron segmentation problem by factorizing the observed datasets into spatial and temporal footprints of the neurons. These methods also take the background fluorescence and noise into account. The choice of factorization framework, constraints and background modelling results in different variants of this approach. In this section we summarize the most relevant ones, focusing specially on the commonly used independent component analysis (ICA) [54] and non-negative matrix factorization approaches [55].

5.1. Independent Component Analysis-Based Approaches

Independent component analysis is a computational approach which separates a multivariate signal into additive sub-components. It defines a generative model for the observed multivariate data. The data variables are assumed to be weighted linear combinations of some unknown latent variables. It also assumes that the combination weights for each latent variable are also unknown. The latent variables are assumed to be non-Gaussian and mutually independent. Therefore, the end goal of ICA is to find a new representation of observed data defined by latent variables which are statistically independent of each other. In the context of calcium imaging, the ICA tries to separate the observed fluorescent stack in independent sources which will (hopefully) represent the fluorescence activity of individual neurons. The underlying assumption is that the fluorescence activities of individual neurons are statistically independent.

Mukamel et al. proposed an ICA framework aided by image segmentation to isolate neurons in calcium image data at scale [56]. Individual frames of a calcium imaging stack are first converted into a column vector by rearranging the frame. Then, principal component analysis (PCA) is used to reduce the dimensionality of the image vector which can be of order 10^5. After PCA, the ICA algorithm is used to extract independent components in the spatio-temporal domain by maximizing the non-Gaussian distribution in the temporal domain. Then the extracted independent components are checked for spatial and temporal sparsity. To increase the independence of found components, the filters for all independent components are first smoothed and then applied to the image data. Local regions with strong response to a filter are thresholded and if one filter results in more than one binarized regions, new filters are created which make sure that one filter results in only one binarized response region. This process verifies that regions with correlated neural activity are disentangled. The performance of the proposed approach was quantified by comparing it to ground truth produced by a human annotator. The authors report an F1 score of 75%. The authors also tested their approach on calcium imaging data acquired from the molecular layer of cerebellar lobules V and VI of anesthetized and awake head-fixed mice. They have not reported comparative performance for this dataset but they did report that the shape of filters extracted by ICA was same as of the Purkinje cells they were targeting. The proposed ICA+image segmentation framework offers an advantage in the form of iterative fine tuning by targeting regions with correlated neural activity but it also carries the inherent weakness of the ICA frameworks, i.e., the ICA framework is built upon the assumption that the signal sources (neurons) are independent, which might not always be the case. It also cannot detect cells that do not spike or differentiate cells that fire synchronously. It also identifies neuronal projections that may not be of interest to the user for network-level analysis.

Patel et al. proposed a semi-automated ICA-based neuron segmentation framework for neuronal networks quantification and single cell dynamics analysis in calcium imaging data [57]. Under this framework, initial estimates for neurons' ROI are generated using the ICA framework proposed by Mukamel et al. [56]. The authors then assume that the generated ROIs might not be a correct approximation of the actual neurons ROIs since they might contain merged ROIs or some ROIs might be missed. Therefore, those ROIs are displayed and the user is then first presented with the option to change the ICA parameters to get a better initial estimate of the ROIs. Then, the user can modify the ROIs, delete the ROIs or manually add more ROIs. The authors have neither provided any quantitative figures for their neuron segmentation framework when compared to manual annotation, nor have they tested the segmentation framework against public benchmarks. Therefore, we cannot verify the robustness of their approach.

Tegtmeier et al. developed an open-source MATLAB package for calcium data processing and behavioural data analysis in mice [27]. The package is intended to be used for calcium data acquired from head-mounted imaging setup and behavioural data acquired from multiple cameras. The author call the package "Calcium Activity Explorer" or CAVE. The package has sub-routines for motion correction, $\Delta F/F$ calculation, cell (neurons) detection (segmentation) and trace analysis. For cell detection, the ICA framework proposed by Mukamel et al. [56] is used to generate initial ROI estimates. Then thresholds are applied to minimum and maximum sizes of the ROI to delete ROIs smaller or larger than an expected size range (30 to 300 pixels). Those ROIs which overlap by more than 30% are merged together. Moreover, those ROIs whose fluorescence is lower than a threshold are deleted and those ROIs whose temporal correlation is greater than 0.8 are merged. Finally, the users can themselves add or delete ROIs manually. CAVE has sub-routines for calcium activity deconvolution and behavioural analysis as well but they are out of the scope of this survey. The authors have tested their package on data acquired from both a cortical region (primary somatosensory cortex, S1) and a hippocampal region (DG). However, they have not provided any quantitative performance figures for their package on cell/neuron detection.

Prada et al. [52] developed an open-source tool for automatic extraction, counting and localization of calcium signals from calcium imaging stacks based on continuous wavelet transform-guided peak detection in calcium signals. The tool needs three parameters: grid window size (WS) which is used to locate calcium activity events, signal to noise ratio (SNR) which defines what part of the ROI is signal and what is the background and signal average threshold (SAT) which is used as a threshold for background removal. The average signal threshold is either entered manually or calculated from a cell-free region/low activity region of the stack. These three parameters are used to extract an automated pdf report about time and location of calcium activity in the imaging stack. This tool was developed in ImajeJ and R. The authors have not provided a comparative or quantitative performance for this tool. This tool is simple to understand, easier to use and according to the authors and performs activity detection satisfactorily. However, this tool is non-robust as it cannot deal with varying levels of background activity. It cannot isolate neurons on its own and relies solely on a fixed background model.

5.2. Non-Negative Matrix Factorization-Based Approaches

In NMF frameworks, the observed data are described by the vector product of two lower rank matrices, one for the spatial footprint and one for temporal footprint along with some additive terms. The spatial and temporal footprints are constrained to be non-negative since neither the spatial coordinates nor the temporal traces of neurons can be negative.

Majee et al. formulated the neuron detection problem as an image reconstruction problem where the reconstructed image encodes the location of neuron centres [58]. The shapes of the neurons are encoded by the impulse response of a filter in the forward model and is estimated from training data. First, the neurons are represented in two groups, sparsely expressed and overly expressed neurons. Both of these groups are represented by two separate images, $X^{(1)}$ and $X^{(2)}$. These images are then estimated using a matrix factorization framework of four elements where the first element represents the matrix product of spatial and temporal footprints of the neurons, the second element represents the background fluorescence, the third element is white Gaussian noise, while the fourth element is noise from the dendrites. $X^{(2)}$ and $X^{(2)}$ are assumed to be independent, so their prior joint probability can be estimated by their individual prior probabilities. These probabilities are proportional to the L1 norms of $X^{(1)}$ and $X^{(2)}$. Taking all of these facts into account, $X^{(1)}$ and $X^{(2)}$ are estimated by a non-negative matrix factorization framework. This approach has been validated on a manually annotated dataset. It achieved the best performance figures with $F1 = 91\%$, $\textit{precision} = 95\%$ and $\textit{recall} = 87\%$. This approach has not been validated on an established benchmark, so its performance against other neuron detection approaches is not established.

Jinghao et al. proposed a modular pipeline for enhancement, movement correction and signal extraction in one-photon calcium imaging stacks [59]. Each sub-module can be used as a stand-alone unit for their intended purposes. The enhancement modules removes grainy noise by applying anisotropic diffusion-based noise removal [60]. This is followed by morphological opening operations to remove smaller ROIs which might not be necessarily neurons. The movement correction is achieved by hierarchical video registration. The neural enhancement and motion correction are pre-processing steps which are followed by two more steps for calcium activity extraction. First, an over-complete set of seeds is initialized which might contain many false-positives. Then, this over-complete set is cleansed by applying a two-component Gaussian mixture model (GMM) on the traces of the seeds. The traces with relatively larger fluctuations are retained while the rest are discarded. To further strengthen the cleansing, an offline-trained long short-term memory (LSTM) network is used as a classifier to identify actual calcium spikes. The overlapping seeds which exhibit similar temporal activity are then merged together to remove redundant seeds. After seed initialization and cleansing, a constrained non-negative matrix factorization (CNMF) proposed by Pnevmatikakis et al. [31] is used to extract spatial and temporal footprint of the neurons. The authors have validated MIN1PIPE on a synthetic video for comparison with other approaches. The authors report a precision of 100% on this video and claim the pipeline successfully avoids almost all of the false positives found by CNMF. Next, the authors validated MIN1PIPE on calcium imaging stacks obtained from the barrel of the cortex in freely behaving mice. MIN1PIPE detected 357 potential ROI as compared to 79 by PCA/ICA and 598 by CNFM. The authors have not reported quantitative results on a ground truth, so we cannot evaluate its performance against a baseline.

Pnevmatkakis et al. proposed a matrix factorization-based approach for calcium signal extraction which tries to solve the problems of denoising, factorization and deconvolution simultaneously [31]. First, a parametric model is estimated for calcium transients under the assumption that sampling time is less than the decay time of the transients. This is achieved by approximating the calcium transients as the impulse response of an auto-regressive (AR) process of order p (p is either 1 or 2). Next, the spiking signal is estimated by solving a non-negative sparse constrained deconvolution problem. To achieve this, a T time steps field of view is considered. This field of view can be considered as $d \times T$ matrix where d is the number of elements in one time step (all the pixels in one image/slice). The number of neurons in this field of view is initially assumed to be K. The spatio-temporal activity of a neuron is assumed to be an AR process with two elements, one element is for the spatial footprint and the other element is for the temporal footprint in $d \times T$ field. The spatial and temporal footprints are extracted by factorizing the field in two lower rank matrices in a constrained way. The first constraint is that the residuals between observed and reconstructed videos should not have higher noise power than a threshold. The second constraint is the spatial and temporal footprints cannot have negative values. The third constraint is that the spatial components need to be sparse. Keeping all these constraints in mind, the overlapping components are merged iteratively. To achieve this, those components whose temporal correlations are higher than a threshold are merged after each iteration. In order to address a situation where the neurons are densely packed and lazy initialization might result in loss of resolutions in terms of lost neurons, the neurons are treated as Gaussian kernels and the initialization seeks to minimize the number of Gaussian kernels which can explain the observed activity. The approach is tested on an in vitro dataset containing 207 spinal motor neurons. The calcium imaging stacks are acquired at 14.6 Hz. The authors report a strong correlation between the activity traces obtained by the proposed approach and raw traces. On a simulated dataset, the proposed approach achieves a correlation of 0.965 between the ground truth and the extracted traces. The authors have not reported the performance of their approach in terms of neuron detection, or on the Neurofinder challenge [61] or any other manually curated dataset.

Ferran et al. proposed a wavelet transform powered image segmentation and sparse dictionary learning-based pipeline for identifying neuronal activity in calcium imaging [62].

They refer to this pipeline as "ADINA". The pipeline is composed of two components, cell sorting and calcium transients extraction. ADINA solves the cell sorting as a sparse dictionary learning problem where a dictionary element represents a group of co-activated cells (basis functions). The $N \times T$ image stack (N is the number of pixels in an image and T is the number of images in the stack) is assumed to be made of two elements, an $N \times K$ spatial element and a $T \times K$ temporal element where K represents the number of dictionary elements. A representative set of sparse basis functions is learned by keeping either one of the components (spatial and temporal) fixed while iterating over the other. After extraction of the basis functions, those basis functions are identified that might represent more than one cell by non-maximal suppression [63]. Finally, the cells are refined by fitting a 2D Gaussian for each cell. The authors validated their approach on both synthetic and real data. On synthetic data, the authors report a sensitivity of 94.3%. On a real imaging stack acquired from hippocampal cultures of different mice, the authors have qualitatively validated their approach but since no ground truth was available, they did not report quantitative results. This approach is easier to use and adapt for imaging stacks acquired under different settings; however, it is not robust against spatially overlapped cells, varying noise levels and spatially variant background fluorescence.

Pnevmatikakis et al. proposed a structured matrix factorization approach to simultaneously detect and segment ROIs, separate overlapping ROIs and deconvolve the calcium activity traces in calcium imaging stacks [64]. They used the standard matrix factorization approach by treating the imaging stacks as the product of two low rank matrices, one matrix represents the spatial footprint of the neurons while the other matrix represents the temporal activity of the neurons. However, they put a hard constraint on the energy of residual signal between the raw data and denoised signal reconstructed from the factorized matrices. These hard threshold noise levels are estimated by exploiting the autoregressive structure of the calcium indicator dynamics in an iterative manner. This enforces the dynamics of the calcium indicator. Moreover, the authors have proposed warm initialization methods for matrix factorization to increase the computational efficiency. As the fluorescence from individual neurons is highly localized, the noise levels and ROI estimation is further facilitated by taking into account the noise levels and ROIs in previous iterations. To merge ROIs which might belong to the same neuron, they construct a graph where each vortex corresponds to a neuron which is connected to other neurons by an edge if those neurons overlap. If they edge is stronger than a threshold, the ROIs are merged. Moreover, if some ROIs do not contribute to the overall spatio-temporal activity, they are removed. The downside of this approach is that the number of initial ROIs must be known. The authors have validated their approach on in vivo mouse V1 spontaneous activity data, however, they have not compared the results to any ground truth, so a quantitative comparison is not done.

Maruyama et al. proposed a constrained matrix factorization-based approach for neuron detection by using the bleach line of background fluorescence to disassociate noise from calcium activity [65]. The matrix factorization approach follows the standard protocols by assuming that the observed calcium imaging data are the product of two low rank matrices, one corresponding to the spatial footprint of the neurons and the other corresponding to the temporal footprint. They also add a background noise term and base fluorescence term which itself is the product of baseline background spatial and temporal activity. All these terms are constrained to be positive. Overall, there are four unknowns which need to be estimated: S which represents the spatial footprint, A which represents the temporal activity, s_b which represents the baseline (background) spatial footprint and a_b which represents the baseline temporal footprint. All the matrices are randomly initialized. In one iteration, a least squares solution is found first for one matrix while the others are fixed and all the negative values are zeroed out. This process is repeated for pre-determined number of iterations. The ROIs detected in this approach are further refined by manual screening and merging. The authors have reported satisfactory performance on the spike deconvolution tasks and temporal activity extraction tasks when applied to both synthetic and real data. However, they have not reported its performance on a ground truth for cell detection, which can help us determine how well this algorithm works on cell/neuron detection tasks.

Friedrich et al. proposed an approach for large-scale neuron detection in whole brain imaging based on non-negative matrix factorization (NMF) applied to patches of the acquired calcium imaging stack, instead of applying the NMF to the whole field of view [66]. The core idea behind this approach is when NMF is applied to a small patch of the stack, instead of the whole stack, the spatial and temporal components can be extracted with higher speed owing to the low complexity of factorization space. Then the method can be applied iteratively to all the patches. This approach has two benefits: one, it can be scaled linearly and two, since it is applied on localized patches, information from two spatially disjoint but visually similar patches will not leak. Moreover, they also argue that spatial and temporal downsampling can reduce speed while not compromising the overall performance, thus further boosting the processing speed. The patches are centred around potential neurons whose centres are extracted by local maxima, therefore this approach is also vulnerable to neuron-like noise. The authors have not provided any quantitative results in order to compare this approach to other state-of-the-art approaches in terms of detection accuracy, precision and recall.

Pachitariu et al. proposed a pipeline for motion-correction, neuron detection and spike inference in calcium imaging. They call the pipeline Suite2p [67]. The pipeline can be scaled linearly with the number of neurons in the field of view. They formulate a generative model of the fluorescence image based upon spike times and neuropil signal. As per their argument, neuropil and soma signals are highly correlated and cell transients can be identified by the deviation from otherwise linear relationship between soma and neuropil. They model recorded signal as a weighted sum of temporally modulated cell signal and temporally modulated neuropil signal at each pixel. They find the parameters of temporal modulation, weights of soma and neuropil and noise parameters by applying expectation maximization on the observed data. They fix some parameters and iterate over the others in one optimization iteration. Once the ROIs are detected iteratively, a user defined threshold for the size of soma is used to identify soma and neuropil. In order to speed up the computations, they use SVD to reduce data dimensions and then apply the pipeline in lower dimensions. On the Neurofinder benchmark, this approach achieved good performance with a precision of 57.8% ± 16%, a recall of 56.8% ± 18% and F1 = 55% ± 12.7%. One big advantage of this pipeline is that it can be scaled linearly with the number of recorded cells. However, since this pipeline relies on accurate parameter estimation for downstream analysis, it can face problems with a region with high cell density. Moreover, its robustness on highly occluded and spatially overlapped cells is also not well established.

Diego et al. approached cell detection, spike inference and impulse response calculation as a single optimization problem in extremely sparse calcium imaging stacks [68]. They assume that all the neurons in the field of view will fire an extremely sparse rate J, much lower than number of cells K. Moreover, each neuron (cell) can be decomposed into the convolution of a number of impulse responses L (again much lower than K). The J, K, L and H (size of convolutional kernel) are specified by the user. All the unknown variables in the optimization problem are first initialized from a non-negative Gaussian distribution. The optimization problem is then solved by fixing a group of variables while iterating over the other variables. The biggest disadvantage of this approach is that the user has to specify at least four parameters which makes this approach vulnerable to human error and makes it harder to be used by a non-expert. The authors have not reported any performance figures for neuron detection from this pipeline.

Giovannucci et al. proposed an online framework for denoising, neuron detection and spike deconvolution which they call Online Analysis of streaming Calcium Imaging Data (OnACID) [69]. This approach is based on online dictionary learning approaches. In this approach, they model the neuron detection as a matrix factorization problem. The observed data are assumed to be a product of spatial footprint and temporal footprint with some noise and background/neuropil contamination added. The background signal is further decomposed into spatial and temporal components and are estimated by non-negative matrix factorization. This matrix factorization regime works well if the number of sources (neurons) are fixed but struggles when more neurons are added/deleted. To overcome this, a new neuron is added if the correlation between the new neuron and current neurons in two successive data frames is lower than a threshold. Moreover, a Gaussian smoothing kernel is applied with a size similar to the size of expected new neuron and then search for point with maximum spatial variance. In order to speed-up the computation, parameter values in an iteration are initialized from the values in previous iterations. Moreover, the neurons statistics are updated only after certain number of steps, thus further reducing the processing time. The authors have validated their approach on both synthetic and real data. On synthetic data, this approach detected and tracked all the active sources with only one false positive. The approach was also bench-marked on data manually annotated by two persons. This dataset consisted of 90k frames acquired at 30 Hz (50 min). The proposed approach achieved precision = 0.87, recall = 0.72 and F1 = 0.79. Moreover, OnACID analysed the dataset in 27 min, suggesting that it can process almost 2 times faster than real time.

Inan et al. proposed a robust statistical optimization method for matrix factorization framework to detect and extract neural cells in calcium imaging ([70]). This optimization framework assumes the observed data are a matrix product of spatial and temporal footprints with background noise. The background noise is dominated by baseline activity modelled by a normal distribution. The cell extraction starts by detecting local maxima in the time maximum summary image, one at a time. Then at each iteration, the matrix factorization framework is solved for spatial and temporal footprints using M-estimator and Huber loss [71,72]. This is repeated until a convergence criteria is satisfied. The redundant components and those components which have converged to zero are removed. The authors have validated their approach on both synthetic and real data. Although they provided a qualitative comparison of their proposed approach against some other matrix factorization frameworks, yet they have not provided any performance figures on a ground truth of established benchmarks, so the performance of this approach on benchmarks is not validated.

Pengchang et al. proposed another variant of constrained non-negative matrix factorization for extracting neurons in calcium imaging [73]. They model the observed data using three elements. The first element is the matrix product of spatial and temporal footprint of the neurons, the second element represents the background fluorescence while the third element is Gaussian noise. The constraints on spatial footprints are that they should be spatially localized and sparse. The temporal footprints are modelled as highly structured auto-regressive processes. Consequently, the combined spatio-temporal neural activity is structured and sparse. The background fluorescence is made of two elements, the baseline fluorescence and a fluctuating component. The fluctuating element at each pixel is estimated by the linear combination of nearby pixels which are not the nearest neighbours. This approach has been validated on both synthetic and real data. It identified all the neurons in a synthetic dataset. In a dorsal striatum dataset, this approach identified 692 neurons against 547 neurons identified by PCA/ICA. The authors have not provided any details about the performance of this approach on established benchmarks.

6. Machine Learning-Based Methods

6.1. ML Methods Applied to Summary Images

In this sub-section, we cover all those ML/DL based approaches which apply an ML/DL algorithm to the summary images of the input stacks. Therefore, the first step of these approaches after preprocessing is always the extraction of an appropriate summary image.

Valmianski et al. [74] proposed a multi-staged classification pipeline to determine which pixel belongs to a neuron or otherwise. The first two stages consist of robust-boost classifiers [75]. The first stage classifier classifies every pixel into neuron and non-neuron classes based on eight statistical features. A median filter is then used to remove small regions falsely identified as neurons. The second stage classifier is another robust-boost classifier which classifies if a candidate neuron is actually a neuron or false positive based on six morphological features. The output probability map is then thresholded and neurons are segmented using connected thresholds. The authors have tested their approach on 64 different datasets, resulting from 16 different regions imaged over 4 trials on 4 mice. Each data set consisted of 200 frames at a resolution of 256 × 256 pixels. They have reported a test error of 7% and ROC curve maxing at 0.97.

Wang et al. [76] proposed a two staged neuron detection pipeline. In the first stage, the raw image stack is first denoised by taking a moving average along temporal axis. Then they normalize the stack using local maxima and minima along temporal axis. After normalization, they generate a reference image by taking maximum value along temporal axis and an average image by taking average value along temporal axis. They binarize the reference image using Local Adaptive Thresholding Algorithm (LATA) [77]. After binarization, they use watershed algorithm on the binary and reference image to create intermediate segmentation mask. To separate neurons from non-neuron blobs in the segmentation mask, they train a three layered CNN. The first and second layers consist of a convolutional layer followed by a max-pooling layer while the third layer is fully connected layer generating two outputs, 1 for neurons and 0 for otherwise. They have evaluated their pipeline on nine manually annotated image stacks and reported F1 score of 85% ± 1%. They have not reported the performance of their pipeline on any benchmarks, so it is hard to compare their pipeline against other state-of-the-art approaches.

Aleksander et al. [78] proposed a standard UNet [21] based neuron segmentation pipeline. They create a single mean summary image by taking average value across temporal dimension. This operation flattens the temporal dimension and converts 3D stack into a 2D image. Then they apply a UNet on 128 × 128 patch of the summary image to segment the neurons. The UNet is composed of an input layer, three encoding layers, one middle layer, three decoding layers and an output layer. The input layer has two 3 × 3 convolutional operations with 32 filters each. First encoding layer consists of two 3 × 3 convolutional layers with 64 filters each and a 0.25 dropout layer. The second encoding layer consists of two 3 × 3 convolutional layers with 128 filters each and a 0.5 dropout layer. The third encoding layer consists of two 3 × 3 convolutional layers with 256 filters each and a 0.5 dropout layer. The middle encoding layer consists of two 3 × 3 convolutional layers with 512 filters each and a 0.5 dropout layer. Each decoding layer consists of concatenation, two 3 × 3 convolutions, a 3 × 3 deconvolution and dropout. The third decoding layer has 256, 256 and 128 filters in each of its two convolutional and one deconvolutional layers, the second decoding layer has 128, 128 and 64 filters in each of its two convolutional and one deconvolutional layers, and the first decoding layer has 64, 64 and 32 filters in each of its two convolutional and one deconvolutional layers.The output layer has a concatenation, two 3 × 3 convolutions with 32 filters each and output softmax layer. They have validated their approach on the Neuronfinder challenge dataset (http://neurofinder.codeneuro.org/, accessed on 2 July 2022). They trained their network on the top 75% of the summary images and validated it on bottom 25%. In 2017, their pipeline stood third on the Neurofinder challenge with an F1 score of 59% ± 16%.

Bao et al. proposed a shallow UNET based active neuron segmentation approach for online use purposes [79]. They apply this approach on single 2D images, which can either be individual images or summary images. First, they do motion correction by homomorphic filtering and large-scale background fluctuation removal. Then the a temporal filter is applied to each pixel to enhance the active pixels and suppress the others. Afterwards, a shallow UNET is trained on active neurons masks. The UNET has two convolution layers and two downsampling/upsampling layers and one skip connection layer. This shallow UNET allows faster processing. The temporal pixel enhancement and faster UNET deployment makes the approach capable of real-time speeds. The authors claim that this approach achieved a processing speed of 117 frames per second on 275 µm ABO dataset. It achieved an F1 = 0.85 on the ABO dataset. This approach is on par with other online segmentation approaches in performance and speed while leveraging the power of both temporal signal processing and machine learning. This approach might still struggle in imaging stacks with high ration of overlapped neurons.

6.2. ML Methods Applied to 3D Calcium Imaging Stacks

In this sub-section, we cover all those ML/DL based approaches which apply an ML/DL algorithm directly to the spatio-temporal stacks. These methods can either be end-to-end or might need an additional post-processing step.

Xu et al. proposed a CNN architecture with a graph regularization term which can utilize unlabelled data along with labelled data for neuron segmentation [80]. First, they over-segment a neuron image into superpixels [81]. The superpixels are then fed to a CNN which minimizes a composite loss function over the labelled and unlabelled superpixels. The CNN is composed of two convolutional layers with 6 and 50 filters each. The loss function over labelled superpixels is represented by a softmax loss while the loss function over unlabelled superpixels is represented by a graph-regularized loss. The graph regularized loss function is formulated on the assumption that neighbouring elements in a graph will most likely share the same label. Therefore, they perform label propagation from labelled superpixels onto unlabelled superpixels by Gaussian field harmonic functions [82]. They repeat the segmentation process for all the images in a volume. Finally, they perform neuron segmentation in 3D volume by global association method [83]. They have validated their approach on a private one-photon data set. They manually annotated 2000 neuron regions spread over 1000 images (which form a 3D volume when stacked up), with half of them used for training and the remaining used for testing. Along with 2000 manually annotated regions, they include 70,000 more unlabelled regions (superpixels) which include 4000 neurons. They have reported an F1 score of 0.96 when they use as few as 40 images for training and an F1 score of 0.99 when they use 1000 images for training.

Apthorpe et al. [84] proposed a 3D (referred to as (2 + 1)D) convolutional network based neuron segmentation approach for calcium imaging data. It accepts an input image stack containing T time slices. There are four 10 × 10 convolutional layers, a max pooling over all time slices and two 1 × 1 bottleneck fully connected layers. The output layer is configured to yield two 2D greyscale images as output, which together represent the softmax probability of each pixel being inside an ROI centroid. The dimensions of the input image stack are $37 \times 37 \times T$, therefore to cover the whole spatial domain, the window is made to slide in two dimensions over the input image stack to produce an output pixel for every location of the window fully contained within the image bounds. The network was trained on two-photon calcium imaging data gathered from both the primary visual cortex (V1) and medial entorhinal cortex (MEC) from awake-behaving mice. size. Human experts annotated ROIs using the ImageJ Cell Magic Wand Tool [85], which automatically generates a ROI based on a single mouse click. The human experts found 4006 neurons in the V1 dataset with an average of 148 neurons per image series and 538 neurons in the MEC dataset with an average of 54 neurons per image series. They have used F1 score to evaluate the network performance and reported F1 = 71%.

Wen et al. [86] proposed a 3D-UNet [21] based neuron segmentation approach on the calcium imaging data acquired from the nematode "Caenorhabiditis elgans". The UNet accepts a 160 × 160 × 16 image stack and produces a binary mask of the same dimension, with every voxel either representing a neuron or background. The 3D UNet consists of three encoding layers, one middle layer and three decoding layers. The first encoding layer consists of two convolutional operations with 8 and 16 filters, second and third also consists of two convolutional operations with 16, 32 and 32, 64 filters. Each encoding layer is followed by 2 × 2 downsampling. The middle layer has 2 convolutional operations with 64 filters each. The output of each encoding layer is concatenated with the input of decoding layer at similar level. The decoding layer consist of one convolutional operation and one concatenation with 8, 16 and 32 filters respectively. The output probability score for each voxel is thresholded at 0.5 and then neurons are segmented by Gaussian blurring and watershed segmentation. The original 3D image stacks are 512 × 1024 × 28, therefore they divided the initial raw images into 160 × 160 × 16 sized smaller images. They have used one 3D stack for training and reported that the pipeline tracked 98% of neurons in the test data. They have also reported that the pipeline was able to track 91% of the neurons even when artificial noise was added.

Peterson et al. [87] proposed a dictionary learning-based approach, called SCALPEL, for neuron detection in calcium imaging data. First, they binarize every frame in the image stack after standardization. Then they use the connected thresholds [88] method to segment candidate neurons from the background. After segmentation, they filter candidate neurons which are smaller or larger than an expected size range. Then they generate a dissimilarity matrix based on temporal and spatial properties of candidate neurons in which each element represents how dissimilar a candidate neuron is compared to another candidate neuron. Using the dissimilarity matrix, they cluster similar candidate neurons together by hierarchical clustering approach [89]. In the final step, they optionally filter the refined dictionary elements, arguing that clusters with a larger number of members are more likely to be true neurons. They have validated their pipeline on three calcium videos/image stacks. The first video is a one-photon video, collected by the lab of Ilana Witten (https://wittenlab.org/, accessed on 2 July 2022) at the Princeton Neuroscience Institute, with 3000 frames of size 205 × 226 pixels sampled at 10 Hz. They have only reported the successful detection of neurons, but no precision, recall or F1 scores. The second and third videos are two-photon videos from Allen Brain Observatory (http://observatory.brain-map.org/visualcoding, accessed on 2 July 2022), containing 105,698 and 105,710 frames respectively of size 512 × 512. Once again, they have reported successful detection of quite large number of neurons but no precision, recall or F1 scores.

Soltanian-Zadeh et al. proposed a spatio-temporal neural network architecture for segmenting active neurons in calcium imaging data [90]. Their architecture is based on DensVNet [91]. Like other popular fully CNNs for semantic segmentation of medical images (e.g., UNet [21] and VNet [92]), DenseVNet is composed of encoding layers, decoding layers and skip connection components. Each encoder stage of DenseVNet is a dense feature stack. The input to each convolutional layer of the stack is the concatenated outputs from all preceding layers of the stack. The authors made the following two modifications to DenseVNet: (i) they changed the last convolutional layer of DenseVNet to have 10 output channels instead of the number of classes and (ii) they added a temporal max-pooling layer to the upsampled features, followed by a 2D convolutional layer with ten 3 × 3 kernels and a final convolutional layer with two 3 × 3 kernels to the output of DenseVNet. They optimized the network using Dice-loss objective function. They validated their pipeline on a subset of the Allen Brain dataset (http://observatory.brain-map.org/visualcoding, accessed on 2 July 2022) and the Neurofinder Challenge dataset (http://neurofinder.codeneuro.org/, accessed on 2 July 2022). The Neurofinder challenge dataset was improved by having it analysed by two expert human annotators. They compared their approach with four other state of the art neuron segmentation approaches namely CaImAn [93], Suite2P [67], HNCcorr [43], and UNet2DS [78]. They have reported a superior true positive detection performance as compared to the states approaches at lower PSNR values; with their approach detecting 80% true positives at PSNR = 10 while the mentioned approaches achieved 65%, 52%, 47% and 41%, respectively. This approach achieved an average F1 score of 69% on the Neurofinder challenge.

Kirschbaum et al. proposed a deep learning powered pipeline for neuron segmentation which explicitly makes use of the temporal and shape information of the neurons [94]. First, the input calcium imaging videos is split in temporal segments. Then, correlations between temporal traces are calculated over all temporal segments. The pixel-wise correlations and a mean summary image is then fed into a CNN which aggregates the information over temporal segments. The outputs of this CNN are fed into a UNet which is then trained to not only produce a foreground background mask, but also produce affinities between neighbouring pixels. The authors have validated their approach on the Neurofinder benchmark and has achieved the best F1 of 67%.

Sita et al. proposed a CNN-based online neuron segmentation and trace extraction pipeline for processing mesoscopic imaging datasets which they call CITE-On (Cell identification and Trace Extraction Online) [95]. This approach is capable of identifying thousands of neurons in real time. The greyscale images are first downsampled, and then upsampled to reduce noise factor and then converted into three channel RGB-like images. These images are then fed into a CNN-based on RetinaNET architecture. The RetinaNET architecture is a single stage detector which uses a focal loss to deal with the class imbalance problem in object detection tasks [96]. The RetinaNET CNN puts bounding boxes around neuron like shapes which are then projected on a summary image. The pixels in each bounding box are then used to create a temporal trace of that neuron. In an online environment, the starting frames are first used to create the initial bounding boxes and traces. When new images become available, the bounding boxes are updated by including the new frames in the imaging stack while removing an equivalent amount of images from the beginning of the imaging stack. The authors claim an average F1 greater than 0.85 for imaging datasets of different calcium indicators, suggesting cross domain generalization capability. In offline settings, the author claim that the performance of CITE-On is on par with STNeuronNet [90] on never before seen datasets.

7. Summary of Performance on Public Benchmarks, Applications and Recommendations

In this section, we provide a performance summary of the approaches mentioned in the previous sections. The performance summary can either be on a public benchmark such as Neurofinder or on a private manually curated dataset. We also provide a link to the online code repositories for these approaches along with a quantitative figure for their performance in the form of precision, recall and detection accuracy. Table 1 summarizes the main methods.

In order to be able to compare the time consumption of the methods surveyed, we also provide a broad approximation of the order of computational complexity at testing time (when available). In the case where a method involves several algorithms, we report the dominant order of complexity among them. Nevertheless, the amount of variables to take into account are closely related to each method. Briefly, in Table 1, we used:

- In the general case, variables W refer to image width, H to image height, D to the depth of the calcium image stack and N to the number of images needed at inference time (some matrix factorization methods need to compute the segmentation using the whole data set).
- For the iterative methods (at inference time) we use m to refer to the number of iterations needed for reaching convergence.
- In the deep learning approaches, we use variable K to denote the average cost of a forward step on a convolutional layer, and F to denote the average cost of a fully connected layer. These costs can be understood as an upper bound to the temporal complexity at testing time, as usually pooling layers reduce the internal feature maps making the pipeline more efficient. The variable F might also be sensible to the size and the number of filters on each convolutional layer.
- We refer with the notation RT to the algorithms where the authors claim that real-time testing performance is achieved.

Most of the ML/DL-based methods are computationally demanding at training time. Usually, these computational needs are accelerated using specific GPU hardware (graphical processing units). Nevertheless, these methods perform faster at testing time. We consider a single pass through the neural network for testing one instance, which is approximated by the addition of the time complexities of the convolutional layers and the fully connected layers present in the network (although filter and layer dimensions from different methods may vary). The capability of performing the training computations only once, completely offline, the use of transfer learning to improve the model estimation and the improved segmentation accuracy at testing time make the ML/DL methods the dominant approach in the most recent literature.

Table 1. Comparison of different approaches. For each method, we provide the reference, a link to the code (if available, accessed on 2 July 2022), the key performance indicators reported (if available), the databases used (MC stands for manually curated and NF stands for the Neurofinder dataset) and the approximate complexity order at testing time.

Type	Code Availability	Performance	DB	Complexity
Guan et al. [44] Signal and image processing-based	MATLAB code available at https://bit.ly/2GItq4O	Precision = 87.5%, Recall = 89.3%, F1 = 88.1%	MC	-
Shen et al. [53] Signal and image processing-based	MATLAB code available at https://bit.ly/328n2Lx	Precision = 71.2%, Recall = 74.9%, F1 = 72.3%	MC	-
Spaen et al. [43] Signal and image processing-based	Python code available at https://bit.ly/36qtJw3	Precision = 65.8%, Recall = 62%, F1 = 59.6%	NF	$O(WHD) + O((WH)^2)$
Reynolds et al. [50] Signal and image processing-based	MATLAB code available at https://bit.ly/3lb776R	F1 = 67.5%	NF	$O(WHK)$ where K is the dimension of the levelset
Giovannucci et al. [69] Matrix factorization	Python code available at https://bit.ly/2SzbJHw	Precision = 87%, Recall = 72%, F1 = 79%	MC	RT
Mukamel et al. [56] Independent component analysis-based	MATLAB code available at https://bit.ly/36PuoaB	F1 = 75%	MC	$O((WHD)^2Nm)$
Pachitariu et al. [97] Matrix factorization-based	Python code available at https://bit.ly/3g55vvT	Precision = 57.8%, Recall = 56.8%, F1 = 55%	NF	-
Pachitariu et al. [98] Matrix factorization-based	Python code available at https://bit.ly/3dSPmai	Precision = 59.9%, Recall = 62.9%, F1 = 58.3%	NF	-
Apthorpe et al. [84] Machine learning-based	Python code available at https://bit.ly/30Ky49C	F1 = 71%	MC	O(4K+2F)
Wang et al. [76] Machine learning-based	MATLAB code available at https://bit.ly/3iEJEcA	F1 = 85%	MC	-
Spaen et al. [43]+Gao et al. [99] Machine learning + signal processing-based	Python code available at https://bit.ly/33EVzTc+ https://bit.ly/36qtJw3	Precision = 70.2%, Recall = 60.2%, F1 = 61.7%	NF	O(6K+F)
Klibisz et al. [78] Machine learning-based	Python code available at https://bit.ly/3d60HTG	Precision = 70.2%, Recall = 60.2%, F1 = 56.9%	NF	O(9K) U-Net 2Ds
Ronneberger et al. [21] Machine learning-based	Python code available at https://bit.ly/36pvvNL	F1 = 69%	NF	O(9K) U-Net
Kirschbaum et al. [100] + [94] Machine learning-based	Python code available at https://bit.ly/30G96YU	F1 = 67%	NF	O(3K+F) CNN +O(5F) U-Net
Soltanian-Zadeh et al. [90] Machine learning-based	Python & MATLAB code available at https://bit.ly/36seu5Q	F1 = 67.5%	NF	O(6K+F)
Bao et al. [79] Machine learning-based	Python & Python code available at https://bit.ly/3wEjsaU	F1 = 85%	NF	-
Sita et al. [95] Machine learning-based	Python & Python code available at https://bit.ly/3sOuE3I	F1 = 85%	MC	-

7.1. Applications and Recommendations

This survey article is intended to function as a set of insights and guidelines to the researchers who want to work with calcium imaging, either in vivo or in vitro. It can be used as a quick preview of the literature on neuron segmentation/neural activity detection in calcium imaging. Additionally, it can be used as a guideline for which method to choose for neuron segmentation in case the researcher gets their hands on a calcium imaging dataset. This survey can be used a guiding set of principle by researchers in the following fields.

- Neuroscientists working on developing semantic neuronal networks from calcium imaging data recorded at sub-cellular resolutions.
- Neuroscientists working to understand the behaviour of neurons affected by some neurodegenerative diseases.
- Researchers working on understanding the firing patterns of neurons as a response to some physical stimuli.
- Researchers working to understand neurons behaviour in detail and then build in silico model of these neurons which in turn can be used to develop neural networks which mimic the behaviour of their biological counterparts.
- Last but not the least, researchers working on neuromorphic computing who want to understand the operations of neuronal networks at sub-cellular resolutions.

Based on the computational requirements, complexity and ease of use, we have following recommendations.

- Signal/image processing-based methods
  - Signal/image processing-based approaches are usually not complex and computationally expensive. A little background in fundamental signal/image processing is enough to get the gist of these methods.
  - In order to research the effect of various parameters on the outcome of these methods, the researchers need to know all the related signal/image processing concepts in detail.
  - For those researchers who are not from a computational background (neuroscientists), the pipeline if these methods is easy to follow and can be used a set of tools with few parameters to tweak.
  - These methods are ideal for neuron segmentation in in vitro datasets since these datasets do not have the overlapping neurons problem as much as the in vivo datasets.
  - These methods tend to be not computationally as expensive, which is why the are also recommended if computational power is an issue. Moreover, without investing a lot on expensive computational hardware, these methods (or at least some of them) can be easily deployed for real-time or online inference.
- Matrix factorization-based methods
  - The base framework of these methods is a bit harder to grasp and can be computationally expensive.
  - For those researchers who are not from a computer science background, these methods might pose a challenge since the success/failure of these methods depend heavily on the choice of factorization framework, background model and baseline fluorescence model.
  - These methods have a modest performance on in vivo datasets as well, so they can be used for neuron segmentation in in vivo datasets with acceptable performance.
  - Since these methods tend to be computationally expensive, they are not recommended if computational cost is an issue. Due to this specific issue, most of these methods are not recommended for real-time or online inference. For instance, we could find only one approach [69] in this category which can be deployed online.
- Machine learning-based methods

- The analytical model behind ML/DL approaches is complex, but ML/DL methods are usually treated as black box models, so if the method is end-to-end (provide data and get a segmentation mask), then these methods are easier to grasp from a holistic perspective.
- These methods perform fairly well on in vivo datasets as well, especially when they are trained on raw calcium imaging stacks and not just on summary images.
- If computational cost is not an issue, then these methods are recommended.
- DL-based methods are always data hungry, so if annotated data availability is an issue, then these methods are not recommended. Some of the DL-based approaches are developed while keeping the online inference in mind such as [79,95] but most of them are data hungry and work well only when deployed offline.

In Table 2 we summarize the main weaknesses and strengths of each category of methods.

Table 2. Comparison of the main strong and weak points of each algorithm category.

Category	Strengths	Weaknesses
Signal/Image processing-based methods	• Simple and explainable • Computational complexity • Fewer parameters to tune • Easily implementable	• Have to compromise either on temporal or spatial information • Difficulties with separating overlapping neurons • Vulnerable to noise • Not transferable • Domain knowledge needed for parameters tuning
Matrix decomposition-based methods	• Better at capturing spatio-temporal context • Explainable • Better performance on in vivo datasets than signal/image processing-based methods	• Computationally expensive • Unwanted artefacts • Not transferable • More problem dependent • Most of these methods cannot be scaled linearly
ML-based methods	• Can be taught to process calcium imaging stacks of different resolutions and frequencies • State-of-the-art results on most of the datasets • Knowledge from one dataset can be transferred to other datasets. • End-to-end learning	• Higher computational power needed in many cases • Require lots of training data • Manual labelling required in most approaches • Mostly black box models. Although researchers have proposed empirical methods for explaining the behaviour of DL models, an analytical explanation is still out of the question.

7.2. Conclusions and Future Work

In this article, we provide a comprehensive overview of the computational methods researchers use for neuron segmentation in calcium imaging. We introduce the problem in the context of spatio-temporal segmentation and then divide the approaches in three main categories: signal/image based approaches, matrix factorization-based approaches and machine learning-based approaches. We describe the core methods shortly and also provide performance figures (if reported by the authors). We also provide a comprehensive list of all the benchmarked approaches and links to their source codes if available. Finally, we also include recommendations for new researchers which are based on the problem definition and the approaches we studied.

Based on the literature survey we conducted, we have identified following areas which can be further improved.

- Neuron segmentation would benefit greatly if more annotated datasets are made available apart from the Neurofinder benchmark.

- In their current form, the annotated datasets of Neurofinder have one two-dimensional mask for a calcium imaging video where only the location of every neuron is identified. It would greatly help if the datasets were annotated in temporal domain as well by adding information about when those neurons were active.
- Machine learning-based approaches can specifically benefit from synthetic data.
- Another interesting avenue to explore is the use of reinforcement learning. We know the firing patterns of neurons and we know they follow the dynamics of calcium indicators. This information can be used to to train reinforcement learning methods to identify neurons.
- Another interesting avenue to explore is generative adversarial networks to learn the calcium imaging distributions and then use the learned distribution for neuron segmentation.
- Since calcium imaging datasets are acquired by different devices and under different conditions, they might not be similar. Moreover, there are other methods as well which capture the same neural activity but in different form (functional magnetic resonance imaging, for example). Therefore, domain alignment and domain adaptation can also help.

Author Contributions: Conceptualization, W.A. and D.M.; methodology, W.A. and D.M.; software, W.A.; formal analysis, W.A. and D.M.; investigation, W.A. and D.M.; resources, W.A. and D.M.; data curation, W.A.; writing—original draft preparation, W.A. and D.M.; writing—review and editing, W.A. and D.M.; visualization, W.A.; supervision, D.M.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the TIN2015-66951-C2-2-R, RTI2018-095232-B-C22 grant from the Spanish Ministry of Science, Innovation and Universities (FEDER funds), and the NVIDIA Hardware grant program.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Research Methods

Appendix A.1. Independent Component Analysis

Independent component analysis (ICA) is a computational method for separating a multivariate signal into additive sub-components. The underlying assumption is that the sub-components are non-Gaussian signals and statistically independent from each other. ICA is a special case of blind source separation. ICA finds the independent components (also called factors, latent variables or sources) by maximizing the statistical independence of the estimated components. There are many ways to choose from in order to formulate the "independence" of the sub-components, thus resulting in different forms of ICA methods. Two sub-components are considered broadly independent if their mutual information is minimized and they are non-Gaussian. In the ICA framework, the observed data are described as the sum of additive statistically independent components. In the case of calcium imaging, the two sub-components are the spatial footprints of the neurons and their temporal traces.

Appendix A.2. Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) is a constrained version of the more general matrix factorization frameworks. In matrix factorization, the observed data are described by decomposing it into two lower rank matrices whose vector product is the closest approximation of the observed data. To take the effects of noise and any other contributing factors into account, more additive terms are added into the framework, depending upon the application and physical quantities being estimated by the framework. In the case of calcium imaging data, the two lower rank matrices represent the spatial and temporal footprints of the neurons. One additive term describes the baseline fluorescence and another additive term describes the noise. Therefore, the general matrix factorization framework for calcium imaging has one lower rank matrix for spatial footprint, one for temporal traces, one term for baseline fluorescence and one for noise. In order to make the framework follow the physics of calcium fluorescence, a constraint is placed so spatial and temporal components are strictly non-negative as neither spatial coordinates nor the temporal traces of neurons can be negative. This is why the framework is often referred to as non-negative matrix factorization (NMF). The additive term for baseline fluorescence can also take many forms, depending on the assumption about the data and the choice of models. Moreover, additional constraints can be put in place and different methods can be chosen to solve the framework which results in a wide range of NMF variants.

Appendix A.3. Summary Images

The image stacks acquired by two-photon calcium imaging cannot be visualized by standard image visualization tools since they have a temporal dimension; therefore, researchers often use summary images. A standard summary image is obtained by applying a specific function across the temporal dimension which collapses the temporal dimension and gives a 2D image representation of the stacks. Based on the collapsing function used, we can divide the summary images into the following two categories.

Appendix A.3.1. Statistical Summary Images

The statistical summary images are obtained when pixels in the 2D image are replaced by a statistical measurement of all the pixels on that specific location across temporal domain [78]. Widely used statistical summary images include the mean image, standard deviation image and max image. For instance, when calculating a mean summary image, the value at a specific spatial location is calculated by taking the mean of all pixels on that specific location across the temporal dimension.

Appendix A.3.2. Correlation Summary Images

The correlation summary images are obtained by finding the mean temporal correlation of a pixel with its neighbours [69]. For instance, to find a pixel value on a specific location in the correlation image, first, the temporal correlation of the pixel on that location is calculated with its immediate neighbours (four or eight neighbours). Then mean of the temporal correlations is placed as the pixel value at that specific location.

References

1. Herculano-Houzel, S. The human brain in numbers: A linearly scaled-up primate brain. *Front. Hum. Neurosci.* **2009**, *3*, 31. [CrossRef]
2. Niedermeyer, E.; da Silva, F.L. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*; Lippincott Williams & Wilkins: Philadelphia, PA, USA, 2005.
3. Bailey, D.L.; Maisey, M.N.; Townsend, D.W.; Valk, P.E. *Positron Emission Tomography*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 2.
4. Ogawa, S.; Lee, T.M.; Kay, A.R.; Tank, D.W. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proc. Natl. Acad. Sci. USA* **1990**, *87*, 9868–9872. [CrossRef]
5. Huettel, S.A.; Song, A.W.; McCarthy, G. *Functional Magnetic Resonance Imaging*; Sinauer Associates: Sunderland, MA, USA, 2004; Volume 1.
6. Stosiek, C.; Garaschuk, O.; Holthoff, K.; Konnerth, A. In vivo two-photon calcium imaging of neuronal networks. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 7319–7324. [CrossRef] [PubMed]
7. Niell, C.M.; Stryker, M.P. Highly selective receptive fields in mouse visual cortex. *J. Neurosci.* **2008**, *28*, 7520–7536. [CrossRef] [PubMed]
8. Ohki, K.; Chung, S.; Ch'ng, Y.H.; Kara, P.; Reid, R.C. Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature* **2005**, *433*, 597–603. [CrossRef]
9. Yaksi, E.; Friedrich, R.W. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca2+ imaging. *Nat. Methods* **2006**, *3*, 377–383. [CrossRef] [PubMed]
10. Najafi, F.; Giovannucci, A.; Wang, S.S.; Medina, J.F. Coding of stimulus strength via analog calcium signals in Purkinje cell dendrites of awake mice. *Elife* **2014**, *3*, e03663. [CrossRef]
11. Gordon, G.; Harville, M.; Woodfill, J.; Darrell, T. Background Estimation and Segmentation Based on Range and Color. U.S. Patent 6,661,918, 9 December 2003.

12. Farin, D.; de With, P.H.; Effelsberg, W. Robust background estimation for complex video sequences. In Proceedings of the 2003 International Conference on Image Processing (Cat. No. 03CH37429), Barcelona, Spain, 14–17 September 2003; Volume 1, p. I-145.
13. Ridder, C.; Munkelt, O.; Kirchner, H. Adaptive background estimation and foreground detection using kalman-filtering. In Proceedings of the International Conference on recent Advances in Mechatronics, Istanbul, Turkey, 14–16 August 1995; pp. 193–199.
14. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2001**, *13*, 556–562.
15. Hoyer, P.O. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* **2004**, *5*, 1457–1469.
16. Mnih, A.; Salakhutdinov, R.R. Probabilistic matrix factorization. *Adv. Neural Inf. Process. Syst.* **2008**, *20*, 1257–1264.
17. Kim, J.; Choi, J.; Yi, J.; Turk, M. Effective representation using ICA for face recognition robust to local distortion and partial occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1977–1981. [PubMed]
18. Lee, T.W.; Lewicki, M.S. Unsupervised image classification, segmentation, and enhancement using ICA mixture models. *IEEE Trans. Image Process.* **2002**, *11*, 270–279. [PubMed]
19. Lu, X.; Wu, H.; Yuan, Y. Double constrained NMF for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2746–2758. [CrossRef]
20. Kayalibay, B.; Jensen, G.; van der Smagt, P. CNN-based segmentation of medical imaging data. *arXiv* **2017**, arXiv:1701.03056.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
22. Cantu, D.A.; Wang, B.; Gongwer, M.W.; He, C.X.; Goel, A.; Suresh, A.; Kourdougli, N.; Arroyo, E.D.; Zeiger, W.; Portera-Cailliau, C. EZcalcium: Open-Source Toolbox for Analysis of Calcium Imaging Data. *Front. Neural Circuits* **2020**, *14*, 25. [CrossRef]
23. Romano, S.A.; Pérez-Schuster, V.; Jouary, A.; Boulanger-Weill, J.; Candeo, A.; Pietri, T.; Sumbre, G. An integrated calcium imaging processing toolbox for the analysis of neuronal population dynamics. *PLoS Comput. Biol.* **2017**, *13*, e1005526. [CrossRef]
24. Denis, J.; Dard, R.F.; Quiroli, E.; Cossart, R.; Picardo, M.A. DeepCINAC: A deep-learning-based Python toolbox for inferring calcium imaging neuronal activity based on movie visualization. *eNeuro* **2020**, *7*. [CrossRef]
25. Pnevmatikakis, E.A. Analysis pipelines for calcium imaging data. *Curr. Opin. Neurobiol.* **2019**, *55*, 15–21. [CrossRef]
26. Stringer, C.; Pachitariu, M. Computational processing of neural recordings from calcium imaging data. *Curr. Opin. Neurobiol.* **2019**, *55*, 22–31. [CrossRef]
27. Tegtmeier, J.; Brosch, M.; Janitzky, K.; Heinze, H.J.; Ohl, F.W.; Lippert, M.T. CAVE: An Open-Source Tool for Combined Analysis of Head-Mounted Calcium Imaging and Behavior in MATLAB. *Front. Neurosci.* **2018**, *12*, 958. [CrossRef]
28. Dubbs, A.; Guevara, J.; Yuste, R. moco: Fast motion correction for calcium imaging. *Front. Neuroinform.* **2016**, *10*, 6. [CrossRef] [PubMed]
29. Mitani, A.; Komiyama, T. Real-time processing of two-photon calcium imaging data including lateral motion artifact correction. *Front. Neuroinform.* **2018**, *12*, 98. [CrossRef] [PubMed]
30. Chen, Z.; Blair, H.T.; Cong, J. LANMC: LSTM-Assisted Non-Rigid Motion Correction on FPGA for Calcium Image Stabilization. In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, CA, USA, 24–26 February 2019; pp. 104–109.
31. Pnevmatikakis, E.A.; Soudry, D.; Gao, Y.; Machado, T.A.; Merel, J.; Pfau, D.; Reardon, T.; Mu, Y.; Lacefield, C.; Yang, W.; et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron* **2016**, *89*, 285–299. [CrossRef] [PubMed]
32. Vogelstein, J.T.; Watson, B.O.; Packer, A.M.; Yuste, R.; Jedynak, B.; Paninski, L. Spike inference from calcium imaging using sequential Monte Carlo methods. *Biophys. J.* **2009**, *97*, 636–655. [CrossRef] [PubMed]
33. Vogelstein, J.T.; Packer, A.M.; Machado, T.A.; Sippy, T.; Babadi, B.; Yuste, R.; Paninski, L. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J. Neurophysiol.* **2010**, *104*, 3691–3704. [CrossRef]
34. Pnevmatikakis, E.A.; Merel, J.; Pakman, A.; Paninski, L. Bayesian spike inference from calcium imaging data. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 349–353.
35. Theis, L.; Berens, P.; Froudarakis, E.; Reimer, J.; Rosón, M.R.; Baden, T.; Euler, T.; Tolias, A.S.; Bethge, M. Benchmarking spike rate inference in population calcium imaging. *Neuron* **2016**, *90*, 471–482. [CrossRef]
36. Speiser, A.; Yan, J.; Archer, E.W.; Buesing, L.; Turaga, S.C.; Macke, J.H. Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4024–4034.
37. Pachitariu, M.; Stringer, C.; Harris, K.D. Robustness of spike deconvolution for neuronal calcium imaging. *J. Neurosci.* **2018**, *38*, 7976–7985. [CrossRef]
38. Liu, T.; Li, G.; Nie, J.; Tarokh, A.; Zhou, X.; Guo, L.; Malicki, J.; Xia, W.; Wong, S.T. An automated method for cell detection in zebrafish. *Neuroinformatics* **2008**, *6*, 5–21. [CrossRef]
39. Kaifosh, P.; Zaremba, J.D.; Danielson, N.B.; Losonczy, A. SIMA: Python software for analysis of dynamic fluorescence imaging data. *Front. Neuroinform.* **2014**, *8*, 80. [CrossRef]
40. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
41. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
42. Mohammed, A.I.; Gritton, H.J.; Tseng, H.a.; Bucklin, M.E.; Yao, Z.; Han, X. An integrative approach for analyzing hundreds of neurons in task performing mice using wide-field calcium imaging. *Sci. Rep.* **2016**, *6*, 20986. [CrossRef] [PubMed]

43. Spaen, Q.; Asín-Achá, R.; Chettih, S.N.; Minderer, M.; Harvey, C.; Hochbaum, D.S. HNCcorr: A novel combinatorial approach for cell identification in calcium-imaging movies. *eNeuro* **2019**, *6*. [CrossRef]
44. Guan, J.; Li, J.; Liang, S.; Li, R.; Li, X.; Shi, X.; Huang, C.; Zhang, J.; Pan, J.; Jia, H.; et al. NeuroSeg: Automated cell detection and segmentation for in vivo two-photon Ca2+ imaging data. *Brain Struct. Funct.* **2018**, *223*, 519–533. [CrossRef]
45. Xu, H.; Lu, C.; Berendt, R.; Jha, N.; Mandal, M. Automatic nuclei detection based on generalized laplacian of gaussian filters. *IEEE J. Biomed. Health Inform.* **2016**, *21*, 826–837. [CrossRef]
46. Xing, F.; Xie, Y.; Yang, L. An automatic learning-based framework for robust nucleus segmentation. *IEEE Trans. Med Imaging* **2015**, *35*, 550–566. [CrossRef]
47. Romano, S.A.; Pérez-Schuster, V.; Jouary, A.; Candeo, A.; Boulanger-Weill, J.; Sumbre, G. A computational toolbox and step-by-step tutorial for the analysis of neuronal population dynamics in calcium imaging data. *bioRxiv* **2017**, 103879. [CrossRef]
48. Tomek, J.; Novak, O.; Syka, J. Two-Photon Processor and SeNeCA: A freely available software package to process data from two-photon calcium imaging at speeds down to several milliseconds per frame. *J. Neurophysiol.* **2013**, *110*, 243–256. [CrossRef]
49. Meyer, F. Topographic distance and watershed lines. *Signal Process.* **1994**, *38*, 113–125. [CrossRef]
50. Reynolds, S.; Abrahamsson, T.; Schuck, R.; Sjöström, P.J.; Schultz, S.R.; Dragotti, P.L. ABLE: An activity-based level set segmentation algorithm for two-photon calcium imaging data. *eNeuro* **2017**, *4*. [CrossRef] [PubMed]
51. Ellefsen, K.L.; Settle, B.; Parker, I.; Smith, I.F. An algorithm for automated detection, localization and measurement of local calcium signals from camera-based imaging. *Cell Calcium* **2014**, *56*, 147–156. [CrossRef] [PubMed]
52. Prada, J.; Sasi, M.; Martin, C.; Jablonka, S.; Dandekar, T.; Blum, R. An open source tool for automatic spatiotemporal assessment of calcium transients and local 'signal-close-to-noise'activity in calcium imaging data. *PLoS Comput. Biol.* **2018**, *14*, e1006054. [CrossRef]
53. Shen, S.P.; Tseng, H.a.; Hansen, K.R.; Wu, R.; Gritton, H.J.; Si, J.; Han, X. Automatic cell segmentation by adaptive thresholding (ACSAT) for large-scale calcium imaging datasets. *eNeuro* **2018**, *5*. [CrossRef]
54. Hyvärinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430. [CrossRef]
55. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef] [PubMed]
56. Mukamel, E.A.; Nimmerjahn, A.; Schnitzer, M.J. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron* **2009**, *63*, 747–760. [CrossRef]
57. Patel, T.P.; Man, K.; Firestein, B.L.; Meaney, D.F. Automated quantification of neuronal networks and single-cell calcium dynamics using calcium imaging. *J. Neurosci. Methods* **2015**, *243*, 26–38. [CrossRef]
58. Majee, S.; Ye, D.H.; Buzzard, G.T.; Bouman, C.A. A Model Based Neuron Detection Approach using Sparse Location Priors. *Electron. Imaging* **2017**, *2017*, 10–17. [CrossRef]
59. Lu, J.; Li, C.; Singh-Alvarado, J.; Zhou, Z.C.; Fröhlich, F.; Mooney, R.; Wang, F. MIN1PIPE: A miniscope 1-photon-based calcium imaging signal extraction pipeline. *Cell Rep.* **2018**, *23*, 3673–3684. [CrossRef]
60. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 629–639. [CrossRef]
61. neurofinder. Available online: http://neurofinder.codeneuro.org/ (accessed on 2 July 2022).
62. Diego, F.; Reichinnek, S.; Both, M.; Hamprecht, F.A. Automated identification of neuronal activity from calcium imaging by sparse dictionary learning. In Proceedings of the 2013 IEEE 10th International Symposium on Biomedical Imaging, San Francisco, CA, USA, 7–11 April 2013; pp. 1058–1061.
63. Blaschko, M.B. Branch and bound strategies for non-maximal suppression in object detection. In Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, St. Petersburg, Russia, 25–27 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 385–398.
64. Pnevmatikakis, E.A.; Gao, Y.; Soudry, D.; Pfau, D.; Lacefield, C.; Poskanzer, K.; Bruno, R.; Yuste, R.; Paninski, L. A structured matrix factorization framework for large scale calcium imaging data analysis. *arXiv* **2014**, arXiv:1409.2903.
65. Maruyama, R.; Maeda, K.; Moroda, H.; Kato, I.; Inoue, M.; Miyakawa, H.; Aonishi, T. Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Netw.* **2014**, *55*, 11–19. [CrossRef] [PubMed]
66. Friedrich, J.; Soudry, D.; Mu, Y.; Freeman, J.; Ahrens, M.; Paninski, L. Fast constrained non-negative matrix factorization for whole-brain calcium imaging data. In Proceedings of the NIPS Workshop on Statistical Methods for Understanding Neural Systems, 2015. Available online: https://www.semanticscholar.org/paper/Fast-Constrained-Non-negative-Matrix-Factorization-Friedrich-Soudry/e4ff845a4b996482f4ef491fff4581a59d949800 (accessed on 2 July 2022).
67. Pachitariu, M.; Stringer, C.; Dipoppa, M.; Schröder, S.; Rossi, L.F.; Dalgleish, H.; Carandini, M.; Harris, K.D. Suite2p: Beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv* **2017**, 061507. [CrossRef]
68. Andilla, F.D.; Hamprecht, F.A. Sparse space-time deconvolution for calcium image analysis. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 64–72.
69. Giovannucci, A.; Friedrich, J.; Kaufman, M.; Churchland, A.; Chklovskii, D.; Paninski, L.; Pnevmatikakis, E.A. Onacid: Online analysis of calcium imaging data in real time. *Adv. Neural Inf. Process. Syst.* **2017**, 30, 2381–2391.
70. Inan, H.; Erdogdu, M.A.; Schnitzer, M. Robust estimation of neural signals in calcium imaging. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2901–2910.
71. Maronna, R.A. Robust M-estimators of multivariate location and scatter. *Ann. Stat.* **1976**, *4*, 51–67. [CrossRef]

72. Huber, P.J. Robust estimation of a location parameter. In *Breakthroughs in Statistics*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 492–518.
73. Zhou, P.; Resendez, S.L.; Rodriguez-Romaguera, J.; Jimenez, J.C.; Neufeld, S.Q.; Giovannucci, A.; Friedrich, J.; Pnevmatikakis, E.A.; Stuber, G.D.; Hen, R.; et al. Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *Elife* **2018**, *7*, e28728. [CrossRef]
74. Valmianski, I.; Shih, A.Y.; Driscoll, J.D.; Matthews, D.W.; Freund, Y.; Kleinfeld, D. Automatic identification of fluorescently labeled brain cells for rapid functional imaging. *J. Neurophysiol.* **2010**, *104*, 1803–1811. [CrossRef]
75. Schapire, R.E.; Freund, Y.; Bartlett, P.; Lee, W.S.; Schapire, R.E. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **1998**, *26*, 1651–1686.
76. Wang, Y.; Su, F.; Wang, S.; Yang, C.; Tian, Y.; Yuan, P.; Liu, X.; Xiong, W.; Zhang, C. Efficient implementation of convolutional neural networks in the data processing of two-photon in vivo imaging. *Bioinformatics* **2019**, *35*, 3208–3210. [CrossRef] [PubMed]
77. Singh, T.R.; Roy, S.; Singh, O.I.; Sinam, T.; Singh, K. A new local adaptive thresholding technique in binarization. *arXiv* **2012**, arXiv:1201.5227.
78. Klibisz, A.; Rose, D.; Eicholtz, M.; Blundon, J.; Zakharenko, S. Fast, simple calcium imaging segmentation with fully convolutional networks. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 285–293.
79. Bao, Y.; Soltanian-Zadeh, S.; Farsiu, S.; Gong, Y. Segmentation of neurons from fluorescence calcium recordings beyond real time. *Nat. Mach. Intell.* **2021**, *3*, 590–600. [CrossRef] [PubMed]
80. Xu, K.; Su, H.; Zhu, J.; Guan, J.S.; Zhang, B. Neuron segmentation based on CNN with semi-supervised regularization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 20–28.
81. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef]
82. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-supervised learning using gaussian fields and harmonic functions. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 912–919.
83. Zhang, L.; Li, Y.; Nevatia, R. Global data association for multi-object tracking using network flows. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
84. Apthorpe, N.; Riordan, A.; Aguilar, R.; Homann, J.; Gu, Y.; Tank, D.; Seung, H.S. Automatic neuron detection in calcium imaging data using convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3270–3278.
85. Noah Apthorpe, N.A. Cell Magic Wand. 2016. Available online: https://github.com/NoahApthorpe/CellMagicWand (accessed on 2 July 2022).
86. Wen, C.; Miura, T.; Fujie, Y.; Teramoto, T.; Ishihara, T.; Kimura, K.D. Deep-learning-based flexible pipeline for segmenting and tracking cells in 3D image time series for whole brain imaging. *bioRxiv* **2018**, 385567. [CrossRef]
87. Petersen, A.; Simon, N.; Witten, D. Scalpel: Extracting neurons from calcium imaging data. *Ann. Appl. Stat.* **2018**, *12*, 2430. [CrossRef]
88. Sonka, M.; Hlavac, V.; Boyle, R. *Image Processing, Analysis, and Machine Vision*; Cengage Learning: Boston, MA, USA, 2014.
89. Hastie, T.; Tibshirani, R.; Friedman, J.; Franklin, J. The elements of statistical learning: Data mining, inference and prediction. *Math. Intell.* **2005**, *27*, 83–85.
90. Soltanian-Zadeh, S.; Sahingur, K.; Blau, S.; Gong, Y.; Farsiu, S. Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 8554–8563. [CrossRef]
91. Gibson, E.; Giganti, F.; Hu, Y.; Bonmati, E.; Bandula, S.; Gurusamy, K.; Davidson, B.; Pereira, S.P.; Clarkson, M.J.; Barratt, D.C. Automatic multi-organ segmentation on abdominal CT with dense v-networks. *IEEE Trans. Med. Imaging* **2018**, *37*, 1822–1834. [CrossRef] [PubMed]
92. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
93. Giovannucci, A.; Friedrich, J.; Gunn, P.; Kalfon, J.; Brown, B.L.; Koay, S.A.; Taxidis, J.; Najafi, F.; Gauthier, J.L.; Zhou, P.; et al. CaImAn an open source tool for scalable calcium imaging data analysis. *Elife* **2019**, *8*, e38173. [CrossRef]
94. Kirschbaum, E.; Bailoni, A.; Hamprecht, F.A. DISCo: Deep Learning, Instance Segmentation, and Correlations for Cell Segmentation in Calcium Imaging. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Virtual, 4–8 October 2020; pp. 151–162.
95. Sità, L.; Brondi, M.; Lagomarsino de Leon Roig, P.; Curreli, S.; Panniello, M.; Vecchia, D.; Fellin, T. A deep-learning approach for online cell identification and trace extraction in functional two-photon calcium imaging. *Nat. Commun.* **2022**, *13*, 1–22. [CrossRef] [PubMed]
96. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
97. Pachitariu, M.; Packer, A.M.; Pettit, N.; Dalgleish, H.; Hausser, M.; Sahani, M. Extracting regions of interest from biological images with convolutional sparse block coding. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1745–1753.
98. Pachitariu, M.; Stringer, C.; Schröder, S.; Dipoppa, M.; Rossi, L.F.; Carandini, M.; Harris, K.D. Sourcery, Suite2p for Neurofinder. 2016. Available online: https://github.com/codeneuro/neurofinder (accessed on 2 July 2022).

99. Gao, S. A CNN Based Neurons Detection Approach. 2016. Available online: https://bit.ly/33EVzTc (accessed on 2 July 2022).
100. Kirschbaum, E. Novel Machine Learning Approaches for Neurophysiological Data Analysis. Ph.D. Thesis, University of Heidelberg, Heidelberg, Germany, 2019.