A deep learning algorithm for 3D cell detection in whole mouse brain image datasets

Adam L. Tyson[1*], **Charly V. Rousseau**[1,2*], **Christian J. Niedworok**[1*], **Sepiedeh Keshavarzi**[1], **Chryssanthi Tsitoura**[1], **Lee Cossell**[1], **Molly Strom**[1], **and Troy W. Margrie**[1,✉]

[1]Sainsbury Wellcome Centre, University College London, 25 Howland Street, London, W1T 4JG , United Kingdom
[2]Present address: Sorbonne Université, Institut du Cerveau - Paris Brain Institute - ICM, Inserm, CNRS, Paris, France
[*]Equal contributions

Understanding the function of the nervous system necessitates mapping the spatial distributions of its constituent cells defined by function, anatomy or gene expression. Recently, developments in tissue preparation and microscopy allow cellular populations to be imaged throughout the entire rodent brain. However, mapping these neurons manually is prone to bias and is often impractically time consuming. Here we present an open-source algorithm for fully automated 3D detection of neuronal somata in mouse whole-brain microscopy images using standard desktop computer hardware. We demonstrate the applicability and power of our approach by mapping the brain-wide locations of large populations of cells labeled with cytoplasmic fluorescent proteins expressed via retrograde trans-synaptic viral infection.

Correspondence: ***t.margrie@ucl.ac.uk***

Introduction

To understand the circuits underlying computations in the brain, it is necessary to map cell types, connections and activity across the entire structure. Advances in labelling (1–3), tissue clearing (4–6) and imaging (7–12) now allow for the meso- and microscopic study of brain structure and function across the rodent brain. Analysis of these whole-brain images has lagged behind the developments in imaging (13). Although there are many relevant commercial and open-source bio-image analysis packages available (14–17), these have traditionally been developed for 2D images or for 3D volumes much smaller than a rodent brain.

In rodent studies, an increasingly common whole-brain image analysis task is the identification of individual, labelled cells across the entire brain. Traditionally, this was carried out manually (18–21), but this approach does not scale to all biological questions, particularly when many cells are labelled. Considering that a mouse brain has around 100 million neurons (22), even if only 0.01% of cells in the brain are labelled, a manual approach becomes impractical for any kind of routine analysis.

There are many methods that work well for identifying labelled cells in serial 2D sections, and subsequently registering the images to reference atlases (23–31). However, 2D analysis can be subject to bias as detected cell numbers can be under, or overestimated depending on sampling in the third dimension. There are now methods for 3D cell detection in whole-brain microscopy images (32–34), but these methods are limited to either nuclear labels, or were only validated in small regions of the brain. Although nuclear labels are much simpler to detect than membrane or cytoplasmic markers (as they have a simple shape and can be approximated as spheres and are far less likely to be overlapping in the image) there are many applications in which a nuclear label is not practical or even useful, as in the case of in vivo functional imaging. Any approach must also be validated throughout the brain, as the signal to noise (SNR) characteristics can vary between brain regions (e.g. sources of noise could be falsely detected as a cell). There does not yet exist a quick method for 3D detection of cells with cytosolic labels that has been validated throughout an entire brain. Meeting this need is a highly desired goal within systems neuroscience.

To overcome the limitations of traditional computer vision, machine learning — and particularly deep learning (35) — has revolutionised the analysis of biomedical and cellular imaging (36). Deep neural networks (DNNs) now represent the state of the art for the majority of image analysis, and have been applied to analyse whole-brain images, to detect cells in 2D (23, 28) or to segment axons (37). However, they have two main disadvantages when it comes to 3D whole brain analysis. Firstly, they require large amounts of manually-annotated training data (e.g. for cell segmentation, this would potentially require the painstaking annotation of hundreds or thousands of cell borders in 3D). Secondly, the complex architecture of DNNs means that for big data (e.g. whole-brain images at cellular resolution), large amounts of computing infrastructure is required to train these networks, and then process the images in a reasonable time frame.

To harness the power of deep learning for 3D identification of labelled cells in whole-brain images, we developed a computational pipeline which uses classical image analysis approaches to detect potentially labelled cells with high sensitivity (cell candidates), at the expense of detecting false positives (i.e. geometrically similar objects). This is then followed by application of a DNN to classify cell candidates as either true cells, or artefacts to be rejected. Harnessing the power of deep learning for object classification rather than cell segmentation at a voxel level speeds up analysis (since there are billions of voxels, but many fewer cell candidates) and simplifies the generation of training data. Rather than annotating cell borders in 3D, cell candidates from the initial step can be further classified by the addition of a single (cell or artefact) label.

Results

To illustrate the problem and to demonstrate the software, whole mouse brain images were acquired following retrograde rabies virus labelling. Viral injections were performed into visual or retrosplenial cortex, causing thousands of cells to be cytoplasmically labelled throughout the brain. Data was acquired using serial two-photon microscopy as previously described (19) (Fig. 1). Briefly, coronal sections are imaged at high-resolution (2 µm x 2 µm x 5 µm voxel size) and stitched to provide a complete coronal section. This is carried out for ten imaging planes after which a microtome removes the most superficial 50 µm of tissue and the process is repeated until the entire brain data set is collected. Light emitted from the specimen is filtered and detected via at least two channels, a primary signal channel containing the fluorescence signal from labelled target neurons and a secondary 'autofluorescence' channel that does not contain target signals but provides anatomical outlines. An example single-plane image is shown in Fig. 2a.

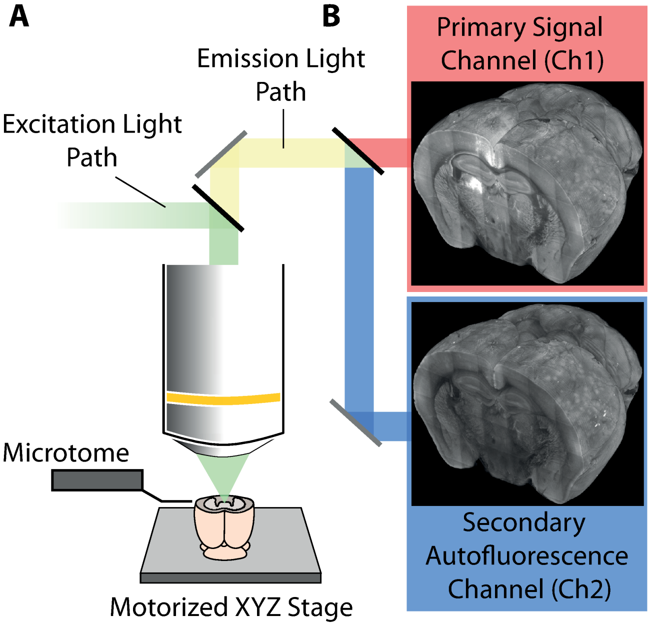

Fig. 1. Simplified schematic diagram of the serial two-photon microscope and data acquisition process. A: The tissue is excited using a femtosecond Ti-sapphire laser (emission wavelength = 800 nm). For data collection, 50 µm of tissue (at approximately 40 µm to 90 µm below the tissue surface) is imaged in ten, 5 µm thick planes. An in-built microtome then physically removes a 50 µm thick section from the optical face. This process is repeated to generate a complete 3D dataset of the specimen. B: For signal collection, the emitted lightpath is split into two channels whereby the primary channel detects the fluorescence signal of interest from labelled cells (e.g. mCherry at 610 nm) and the second channel (e.g. at 450 nm) detects the tissue autofluorescence signal that reveals gross anatomical structure.

Cell candidate detection. When developing any object detection algorithm, a balance must be struck between false positives and false negatives. In traditional (two-dimensional) histology, simple thresholding (e.g. (38)) can often work well for cell detection. This does not necessarily apply to whole brain images. In samples with bright, non-cellular structures (artefacts, Fig. 2c) or lower signal to noise ratio, simple thresholding can detect many non-cellular elements. Image preprocessing and subsequent curation of detected objects can overcome some of these issues, but no single method works reliably across the brain in multiple samples. Either some cells are missed (false negatives), or many artefacts are also detected (false positives). To overcome this, a traditional image analysis approach was used to detect cell candidates, i.e. objects of approximately the correct brightness and size to be a cell. This list of candidates is then later refined by the deep learning step. Crucially, this refinement allows the traditional analysis to produce many false positives while minimising the number of false negatives. Images are median filtered and a Laplacian of Gaussian filter is used to enhance cell-like objects. The resulting image is thresholded, and objects of approximately the correct size are saved as candidate cell positions (Fig. 2d). An overview of the cell candidate detection steps is shown in Fig. S1. The thresholding is tuned to pick up every detectable cell but this also results in the detection of many false positives that often appear as debris on the surface of the brain and in some cases unidentified objects within blood vessels. This initial detection step is based on a number of tunable parameters (Table 5) which were chosen based on trial and error to be biased to over detection, and so be relatively robust to changes in the input data. They may of course need to be tuned for very different data, in particular the in-plane cell somata diameter which may need to be tuned when different types of cells are labelled.

Cell candidate classification using deep learning. A classification step, which uses a 3D adaptation of the ResNet (39) convolutional neural network (Figs. S2 & S3) is then used to separate true from false positives. To classify cell candidates, a subset of cell candidate positions were manually annotated (e.g. Fig. 2e). In total, ~100,000 cell candidates (50,653 cells and 56,902 non-cells) were labelled from five brains. Small cuboids of 50 x 50 x 100 µm around each candidate were extracted from the primary signal channel along with the corresponding cuboid in the secondary autofluorescence channel (Fig. 3a). This allows the network to "learn" the difference between neuron-based signals (only present in the primary signal channel), and other non-neuronal sources of fluorescence (potentially present in both channels).

The trained classification network is then applied to classify the cell candidates from the initial detection step (Fig. 2f). The artefacts (such as those at the surface of the brain and in vessels) have been correctly rejected, while correctly classifying the labelled cells. To quantify the performance of the classification network, and to assess how much training data is required for good performance, the manually annotated training data was split up into a new training dataset from four brains, and a test dataset from the fifth brain. A new network was trained on subsets of the training data, and performance tested on the fifth brain (15,872 cells and 18,168 non-cells). Fig. 3b shows that relatively little training data was required for good performance on unseen test data, with 95% of cell candidates classified correctly with ~7,000 annotated cell candidates. Although ~7,000 data points are required to train the network from scratch, we provide the

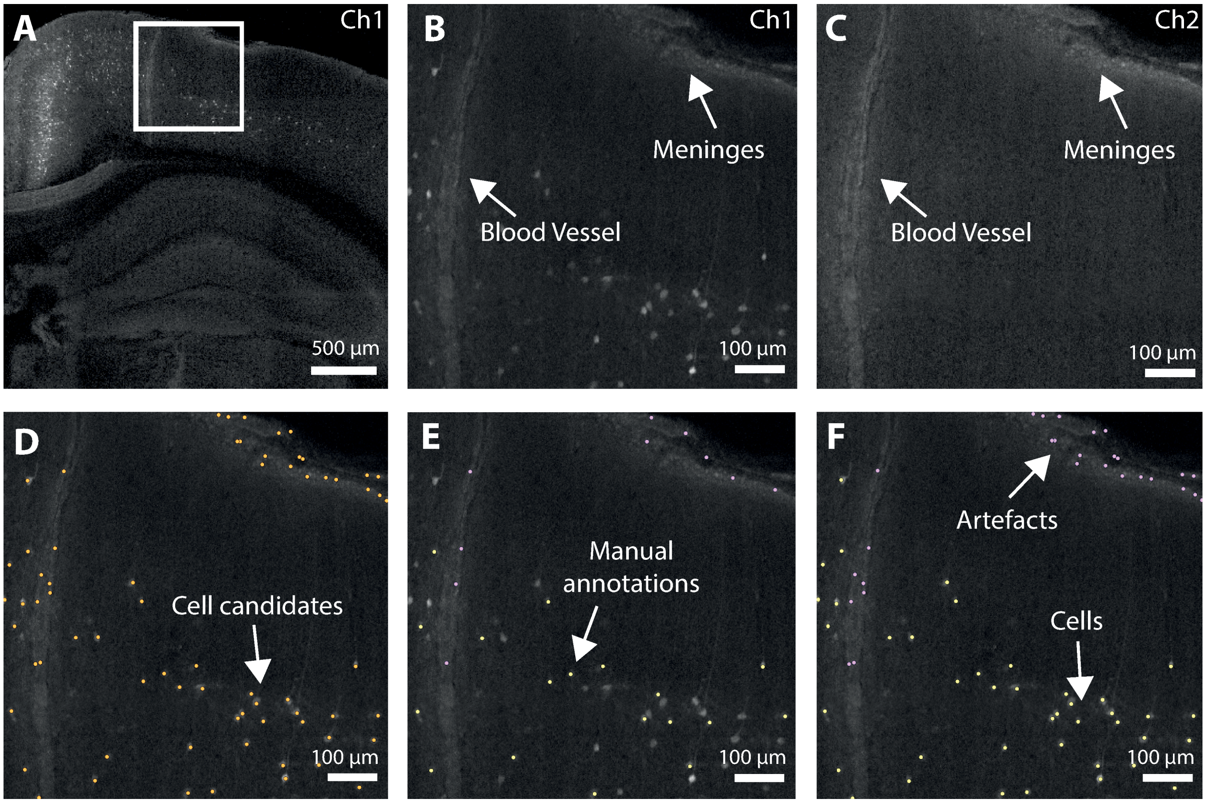

Fig. 2. Illustration of the cell detection process. A: Single coronal plane of raw data (primary signal channel, Ch1). B: Enlarged insert of cortical region from A showing examples of structural features (artefacts) often erroneously detected. C: Cortical region shown in B, in the secondary autofluorescence channel (Ch2). Cells can only be seen in Ch1, but artefacts are visible in both channels. D: Detected cell candidates overlaid on raw data. Labelled cells as well as numerous artefacts are detected. E: Illustration of training data. A subset of detected cell candidates are classified as cells (yellow) or artefacts (purple). Cuboids of data centered on these selected cell candidates are then used to train the network. F: Classified cell candidates. The trained cell classification network is applied to all the cell candidates from (E) and correctly rejects the initial false positives.

network trained on the full dataset with the software. Users can then re-train this network with a much smaller amount of experiment-specific training data.

Application. To illustrate the method, the cell detection software was applied to data which was not used to develop or train the classification network. Data from a previous experiment (40) was acquired on a different microscope and was used to simulate real-world usage in which the SNR characteristics of the data may vary from that used to pre-train the supplied network. Neurons presynaptic to layer 2/3 primary visual cortical cells were labeled using rabies virus tracing (expressing mCherry) in two Penk-Cre mice.

The algorithm was run on these two brains, using the default candidate detection parameters (Table 5). A small number (619) of the detected cell candidates were manually annotated on a single brain (brain 1) including confirmations of correct classifications, and corrections of incorrect classifications. The pre-trained network was then retrained using these data points and 10% of the data was held back for validation during training. The network was trained until the validation loss function began to plateau, taking 73 minutes. The full cell detection algorithm was then repeated for both brains using the re-trained network on a laptop computer. Total time for cell detection was 83 minutes for brain 1 and 91 minutes for brain 2 (for full timings see Table 1).

To assign detected cells to a brain region, the Allen Mouse Brain Reference Atlas (ARA (41)) annotations were registered to the secondary autofluorescence channel using brainreg (42), a Python port of the validated aMAP pipeline (43). These annotations were overlaid on the raw data (Fig. 4a), and the number of cells in each brain region were reported, allowing for quantitative analysis (Table 2).

The new data was noisier than that used to pre-train the network (possibly due to the use of resonant vs galvanometer scanning), which lead to false-positives throughout the brain. A small amount of new training data, taking approximately five minutes to generate was sufficient to significantly improve performance of the algorithm (Fig. 4b). The re-trained network removed many of the false-positives, while still correctly classifying the labelled cells.

Table 1. Algorithm timings on a laptop computer with Intel i9-9900K CPU, 32GB RAM and an NVIDIA RTX2080 GPU. Data stored on an external solid-state drive.

	Brain 1	Brain 2
Number of cell candidates	63995	80122
Number of candidates classified as cells	4266	4021
Time for cell candidate detection	50 minutes	48 minutes
Time for classification	33 minutes	43 minutes
Total time	83 minutes	91 minutes

Validation. To quantify the accuracy of the algorithm brain-wide, we generated ground truth data for both the brain used to generate data for re-training (brain 1) and the “unseen”

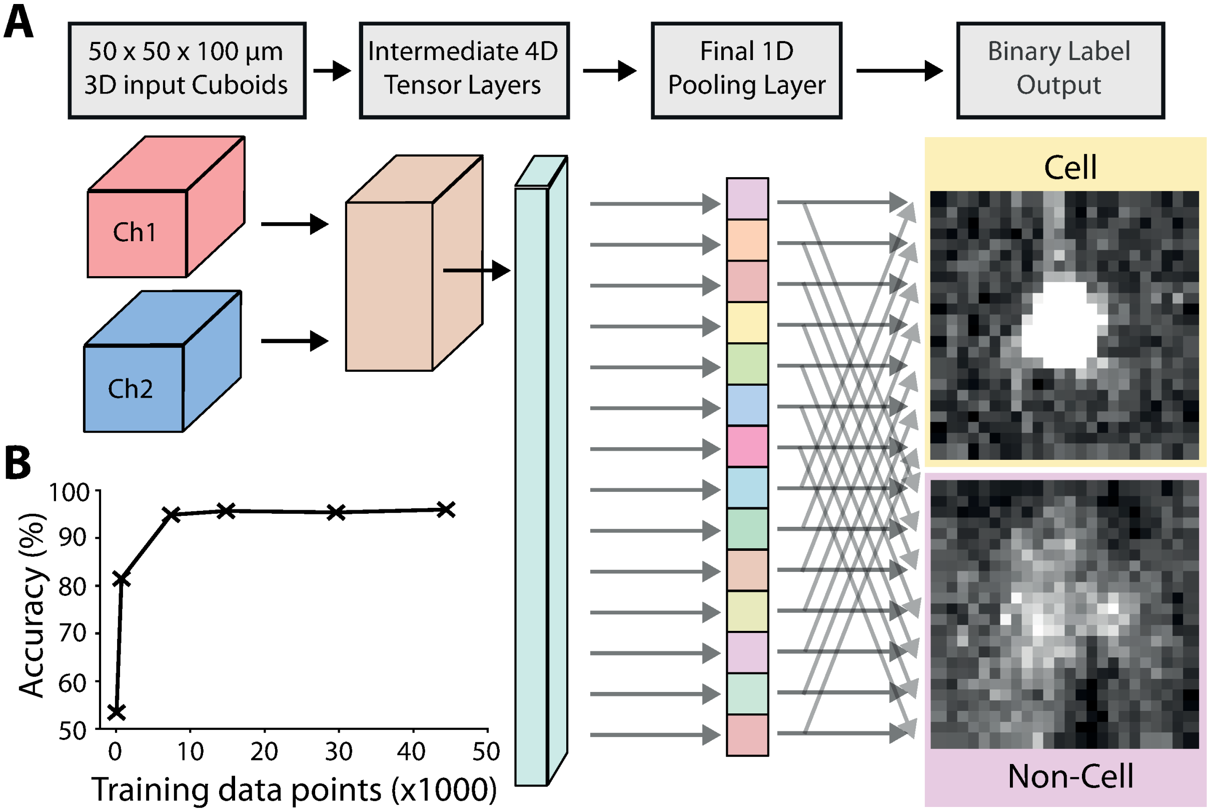

Fig. 3. Cell classification. A: The input data to the modified ResNet are 3D image cuboids (50 µm x 50 µm x 100 µm) centered on each cell candidate. There are two cuboids, one from the primary signal channel, and one from the secondary autofluorescence channel. The data is then fed through the network, resulting in a binary label of cell or non-cell. During training the network "learns" to distinguish true cells, from other bright non-cellular objects. See Figs. S2 & S3 for more details of the 3D ResNet architecture. B: Training the initial cell classification network: classification accuracy as a function of training data quantity.

Table 2. Total number of cells in each region, per brain, projecting to layer 2/3 neurons in primary visual cortex. Ten regions with the greatest number of cells across both brains shown.

Brain structure name	Brain 1	Brain 2
Primary visual area, layer 2/3	965	1222
Primary visual area, layer 5	650	558
Dorsal part of the lateral geniculate complex, core	371	340
Lateral posterior nucleus of the thalamus	240	215
Primary visual area, layer 4	207	205
Retrosplenial area, ventral part, layer 5	162	135
Dorsal part of the lateral geniculate complex, shell	122	124
Lateral dorsal nucleus of thalamus	122	107
Retrosplenial area, dorsal part, layer 5	110	103
Posteromedial visual area, layer 5	46	74

brain (brain 2). Two experts manually annotated all labelled cell somata throughout the brains. These cells were assigned to regions in the ARA in the same way as the automated cell counts and an average of the two experts was taken. The comparison between the automated counts and the manual counts is shown in Fig. 4c. Using the pre-trained network, the algorithm detects false positives, including in many areas with no labelled cells. Re-training the network significantly reduces the number of false positives, bringing the best-fit line closer to an exact match to the manual cell counts. For both brains, a linear fit to the algorithm and manual cell counts is above 1 (brain 1 - 1.113, brain 2 - 1.053), suggesting a small number of false positives (Table 3).

Although the results from the two brains look similar when the detected cells are warped to the ARA coordinate space (Fig. 4d), there is still significant biological variability. For this reason, most experiments quantify relative, rather than absolute, cell counts (18, 20, 24, 44). It is therefore important than the correlation between the automated cell counts and ground truth is as high as possible. The correlation for both the brain used for training (brain 1), and the "unseen" brain (brain 2) is very high (Pearson correlation coefficient, $\rho = 0.999$), and higher than the correlation between the automated cell counts for both brains ($\rho = 0.982$).

Table 3. Comparison between algorithm and expert cell counting.

	Pearson correlation coefficient	Linear best-fit slope
Brain 1		
Algorithm vs expert 1	0.999	1.054
Algorithm vs expert 2	0.999	1.178
Expert 1 vs expert 2	0.999	1.116
Algorithm vs expert mean	0.999	1.113
Brain 2		
Algorithm vs expert 1	0.998	0.964
Algorithm vs expert 2	0.997	1.151
Expert 1 vs expert 2	0.994	1.188
Algorithm vs expert mean	0.999	1.053

Effect of varying axial sampling. All the data presented was acquired with high axial sampling (5 µm), but this is not always possible or desirable due to imaging time, data stor-

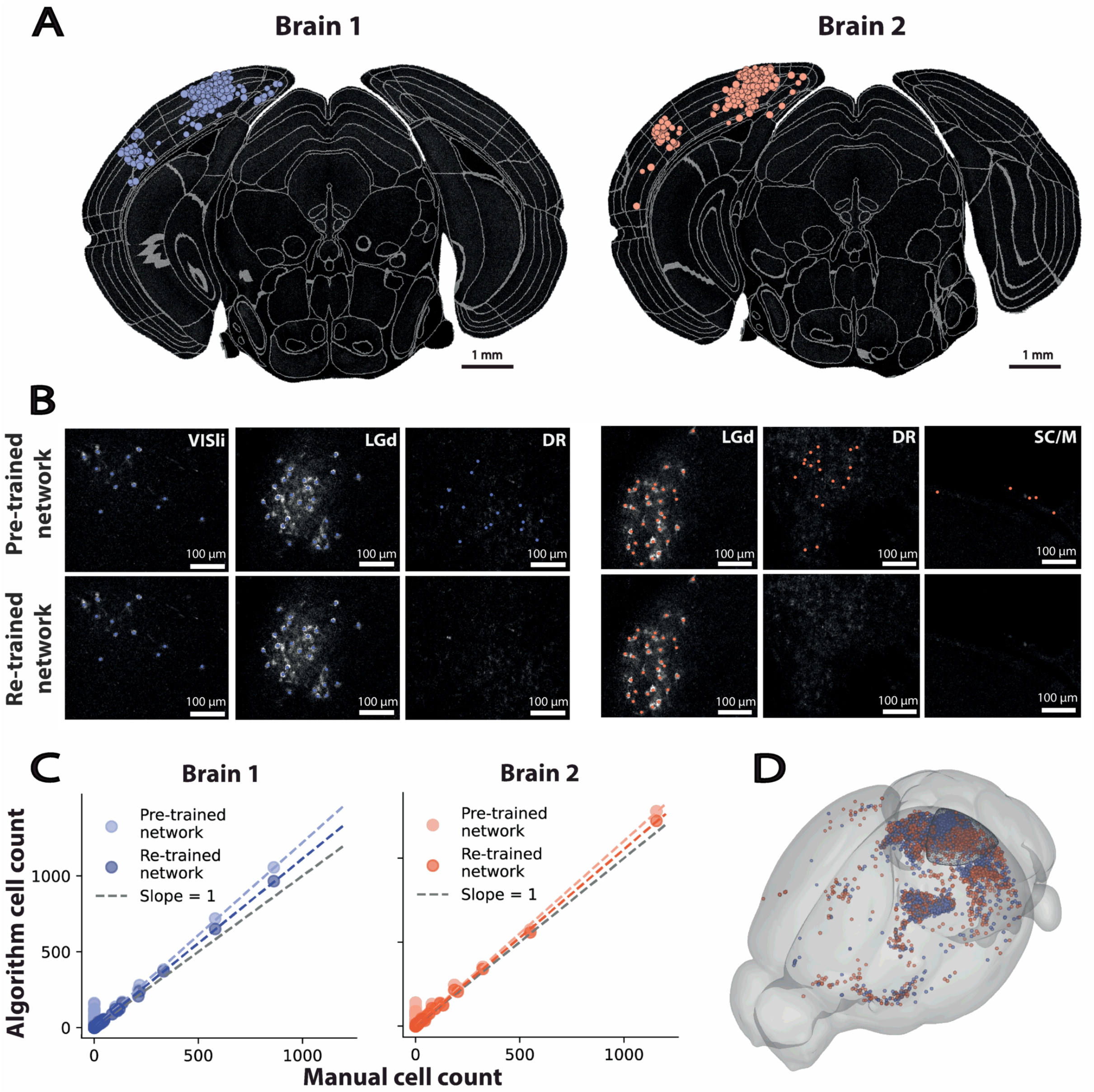

Fig. 4. Application of the algorithm to unseen data. Presynaptic neurons labelled by rabies viral injection into layer 2/3 primary visual cortex in Penk-Cre mice. A: Detected cells overlaid on raw data along with the brain region segmentation for Brain 1 (blue) and 2 (orange). The size of the coloured disk represents the proximity of the cell centroid to the image plane displayed. B: Comparison of cell detection before and after re-training the pre-trained network in different regions of the image. Cells with different morphologies are correctly detected in both dense and sparse regions, and artefacts are rejected. VISli — Laterointermediate area, LGd — Dorsal part of the lateral geniculate complex, DR — Dorsal nucleus raphe, SC/M — Superior colliculus & meninges. C: Comparison of cell counts per ARA brain region between the algorithm and the mean of the two expert counts. Best fit shown before and after re-training. D: Visualisation of detected cells from both brains warped to the ARA coordinate space in 3D, along with the rabies virus injection site target (primary visual cortex, wireframe).

age requirements, or biological considerations such as photobleaching. To assess the performance of the algorithm with varying optical slice thicknesses, we downsampled the data from brain 2, to generate synthetic datasets with axial sampling of 5, 10, 20 and 40 µm, and the algorithm was applied as before.

The cell counts were compared to the mean of expert counts (Table 4). Performance in terms of absolute cell numbers (best fit line slope, and Pearson correlation coefficient) was comparable for 5, 10 and 20 µm, although the 5 µm dataset was most highly correlated to the ground truth. The performance on the 40 µm dataset was much worse. The best-fit line slope of 0.675 means that many cells have been missed, although the high correlation ($\rho = 0.942$) suggests that this effect is relatively uniform throughout the brain.

The results are not surprising, as even with 20 µm axial sampling, most cells can still be visualised in multiple image planes, and so 3D cell detection is still possible. At 40 µm axial sampling, this is not possible, and so many cells are missed. Although the correlation coefficients at 10 and 20 µm axial spacing are relatively high ($\rho = 0.978, 0.981$), they are lower than the correlation between different brains ($\rho = 0.982$). This may reduce the likelihood of detecting small biological effects using data with low axial sampling. There are also many other factors that affect how many image planes a cell will appear in, such as the cell somata size, and the axial point spread function of the microscope.

Table 4. Algorithm performance compared to mean of expert manual counts with varying axial sampling.

Axial sampling [µm]	Pearson correlation coefficient	Linear best-fit slope
5	0.999	1.053
10	0.978	0.906
20	0.981	1.011
40	0.942	0.675

Discussion

Mapping the distribution of labelled neurons across the brain is critical for a complete understanding of the information pathways that underlie brain function. Many existing methods for cell detection in whole-brain images rely on classical image processing, which can be affected by noise, and may not detect complex cell morphologies. DNNs can be used for highly sensitive image processing, but often require laborious generation of training data and are prohibitively slow for the analysis of large, 3D images. The presented method here overcomes these limitations by combining traditional image processing methods for speed, with a DNN to improve accuracy.

Recent developments in microscopy technology (e.g. (12)) now allow for quicker, more routine acquisition of whole-brain datasets. It is important that the image analysis can be carried out in a timely fashion, and without relying on large-scale computing infrastructure. Processing time for the ∼180GB images in Fig. 4 on a laptop was around 90 minutes, so sixteen datasets could be analysed in a single day, much quicker than the sample preparation and imaging steps. Once parameters are optimised, and the classification network is trained, the software can run entirely without user intervention.

In traditional DNN approaches for image analysis, generation of training data is often a major bottleneck. While large-scale "citizen science" approaches can be used to generate large amounts of training data (45), this is not practical for the majority of applications, e.g. when anatomical expertise is required. Our method overcomes this by requiring only a binary label (cell or non-cell) for each cell candidate in the training dataset, rather than a painstaking 3D outline of each cell. The software is released with a pre-trained network, and so the network can be re-trained for specific datasets very quickly. The total time to generate the new network used to classify images in Fig. 4 was less than two hours, including generating training data and retraining the network.

We show that the results of the proposed method compare well to expert manual counts, particularly the correlation between counts in different brain areas. However our results also show the importance of re-training the pre-trained network for new datasets (even if they are superficially similar). It is important to also bear in mind that although we quantified performance of the algorithm across the brain, we did not label every type of cell, and some areas of the brain with densely packed neurons (e.g. hippocampus) were only sparsely labelled. When applying this method to very different data, users should ensure that they re-train the model with representative training data (including different brain areas, cell types and image artefacts if applicable), and check the results in detail. The data in Table 4 shows that although the method is relatively robust to the axial sampling, true 3D imaging (i.e. labelled cells appear in at least two axial planes) is required for accurate cell detection using our method.

The ability to quickly detect, visualise and analyse cytoplasmically labelled cells across the mouse brain brings a number of advantages over existing methods. Analysing an entire brain rather than 2D sections has the potential to detect many more cells, increasing the statistical power and the likelihood of finding novel results, particularly when studying rare cell types. Whole-brain analysis may also be less biased than analysing a series of 2D planes, especially in regions with low cell densities, or differing cell sizes.

This software is fully open-source, and has been written with further development and collaboration in mind. In future we aim to adapt the network to be flexible as to the number of input channels, and output labels. The classification network currently relies on using both the primary signal and the secondary autofluroescence channel. In the future it would be valuable to train a network that could achieve a similar level of performance using a single input channel. Analysing a single channel would allow half as much data to be collected (although autofluorescence channels are optimal for atlas registration). Training a network to produce multiple labels (rather than just cell or non-cell) would allow for cell-type classification based on morphology, or based on gene or protein expression levels if additional signal channels were supplied. Although the ResNet architecture was chosen based on performance (39) and flexibility in new contexts (e.g. (46)), there are many newer network architectures that could be implemented to improve performance (e.g. (47, 48)). Lastly, although this approach was designed for fast analysis of large whole-brain datasets, the proposed two-step approach could be used for any kind of large-scale 3D object detection.

DATA AVAILABILITY

The methods outlined in this manuscript are available within the cellfinder software, part of the BrainGlobe suite of computational neuroanatomy tools. The software is open-source, written in Python 3 and runs on standard desktop computing hardware (although a CUDA compatible GPU allows for a considerable reduction in processing time). Source code is available at github.com/brainglobe/cellfinder and pre-built wheels at pypi.org/project/cellfinder. Documentation, tutorials, and the data underlying Fig. 4 are available at docs.brainglobe.info/cellfinder.

ACKNOWLEDGEMENTS

This work was supported by grants from the Gatsby Charitable Foundation (GAT3361) and Wellcome Trust (090843/F/09/Z and 214333/Z/18/Z) to T.W.M..
This manuscript was typset using a modified version of the HenriquesLab bioRxiv template.

Bibliography

1. Ian R. Wickersham, Stefan Finke, Karl Klaus Conzelmann, and Edward M. Callaway. Retrograde neuronal tracing with a deletion-mutant rabies virus. *Nature Methods*, 4(1):47–49, 2007.
2. Leon G. Reijmers, Brian L. Perkins, Naoki Matsuo, and Mark Mayford. Localization of a stable neural correlate of associative memory. *Science*, 317(5842):1230–1233, 2007.
3. Sung Kim, Jae Hun Cho, Evan Murray, Naveed Bakh, Heejin Choi, Kimberly Ohn, Luzdary Ruelas, Austin Hubbert, Meg McCue, Sara L. Vassallo, Philipp J. Keller, and Kwanghun Chung. Stochastic electrotransport selectively enhances the transport of highly electromobile molecules. *Proceedings of the National Academy of Sciences*, pages E6274–E6283, 2015.
4. Ali Ertürk, Klaus Becker, Nina Jährling, Christoph P Mauch, Caroline D Hojer, Jackson G Egen, Farida Hellal, Frank Bradke, Morgan Sheng, and Hans-Ulrich Dodt. Three-

dimensional imaging of solvent-cleared organs using 3DISCO. *Nature Protocols*, 7(11): 1983–1995, 2012.

5. Kwanghun Chung, Jenelle Wallace, SungYon Kim, Sandhiya Kalyanasundaram, Aaron S Andalman, Thomas J Davidson, Julie J Mirzabekov, Kelly A Zalocusky, Joanna Mattis, Aleksandra K Denisin, Sally Pak, Hannah Bernstein, Charu Ramakrishnan, Logan Grosenick, Viviana Gradinaru, and Karl Deisseroth. Structural and molecular interrogation of intact biological systems. *Nature*, 497(7449):332–337, 2013.
6. Etsuo A. Susaki, Kazuki Tainaka, Dimitri Perrin, Fumiaki Kishino, Takehiro Tawara, Tomonobu M. Watanabe, Chihiro Yokoyama, Hirotaka Onoe, Megumi Eguchi, Shun Yamaguchi, Takaya Abe, Hiroshi Kiyonari, Yoshihiro Shimizu, Atsushi Miyawaki, Hideo Yokota, and Hiroki R. Ueda. Whole-Brain Imaging with Single-Cell Resolution Using Chemical Cocktails and Computational Analysis. *Cell*, 157(3):726–739, 2014.
7. HU Dodt, Ulrich Leischner, and Anja Schierloh. Ultramicroscopy: three-dimensional visualization of neuronal networks in the whole mouse brain. *Nature Methods*, 4(4):331–336, 2007.
8. Timothy Ragan, Lolahon R Kadiri, Kannan Umadevi Venkataraju, Karsten Bahlmann, Jason Sutin, Julian Taranda, Ignacio Arganda-Carreras, Yongsoo Kim, H Sebastian Seung, and Pavel Osten. Serial two-photon tomography for automated ex vivo mouse brain imaging. *Nature Methods*, 9(3):255–258, 2012.
9. Pavel Osten and Troy W. Margrie. Mapping brain circuitry with a light microscope. *Nature Methods*, 10(6):515–523, 2013.
10. Qinyi Fu, Benjamin L. Martin, David Q. Matus, and Liang Gao. Imaging multicellular specimens with real-time optimized tiling light-sheet selective plane illumination microscopy. *Nature Communications*, 7:11088, 2016.
11. Kaoru Seiriki, Atsushi Kasai, Takeshi Hashimoto, Wiebke Schulze, Misaki Niu, Shun Yamaguchi, Takanobu Nakazawa, Ken ichi Inoue, Shiori Uezono, Masahiko Takada, Yuichiro Naka, Hisato Igarashi, Masato Tanuma, James A. Waschek, Yukio Ago, Kenji F. Tanaka, Atsuko Hayata-Takano, Kazuki Nagayasu, Norihito Shintani, Ryota Hashimoto, Yasuto Kunii, Mizuki Hino, Junya Matsumoto, Hirooki Yabe, Takeharu Nagai, Katsumasa Fujita, Toshio Matsuda, Kazuhiro Takuma, Akemichi Baba, and Hitoshi Hashimoto. High-Speed and Scalable Whole-Brain Imaging in Rodents and Primates. *Neuron*, 94(6):1085–1100.e6, 2017.
12. Fabian F. Voigt, Daniel Kirschenbaum, Evgenia Platonova, Stéphane Pagès, Robert A. A. Campbell, Rahel Kastli, Martina Schaettin, Ladan Egolf, Alexander van der Bourg, Philipp Bethge, Karen Haenraets, Noémie Frézel, Thomas Topilko, Paola Perin, Daniel Hillier, Sven Hildebrand, Anna Schueth, Alard Roebroeck, Botond Roska, Esther T. Stoeckli, Roberto Pizzala, Nicolas Renier, Hanns Ulrich Zeilhofer, Theofanis Karayannis, Urs Ziegler, Laura Batti, Anthony Holtmaat, Christian Lüscher, Adriano Aguzzi, and Fritjof Helmchen. The mesoSPIM initiative: open-source light-sheet microscopes for imaging cleared tissue. *Nature Methods*, 16:1105–1106, 2019.
13. Adam L Tyson and Troy W Margrie. Mesoscale microscopy for micromammals: image analysis tools for understanding the rodent brain. *arXiv*, (2102.11812), 2021.
14. Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, jul 2012. ISSN 1548-7105.
15. Fabrice De Chaumont, Stéphane Dallongeville, Nicolas Chenouard, Nicolas Hervé, Sorin Pop, Thomas Provoost, Vannary Meas-Yedid, Praveen Pankajakshan, Timothée Lecomte, Yoann Le Montagner, Thibault Lagache, Alexandre Dufour, and Jean Christophe Olivo-Marin. Icy: An open bioimage informatics platform for extended reproducible research. *Nature Methods*, 9(7):690–696, 2012.
16. Claire McQuin, Allen Goodman, Vasiliy Chernyshev, Lee Kamentsky, Beth A. Cimini, Kyle W. Karhohs, Minh Doan, Liya Ding, Susanne M. Rafelski, Derek Thirstrup, Winfried Wiegraebe, Shantanu Singh, Tim Becker, Juan C. Caicedo, and Anne E. Carpenter. CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biology*, 16(7):1–17, 2018.
17. Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. Ilastik: Interactive Machine Learning for (Bio)Image Analysis. *Nature Methods*, 16:1226–1232, 2019.
18. Mitsuko Watabe-Uchida, Lisa Zhu, Sachie K. Ogawa, Archana Vamanrao, and Naoshige Uchida. Whole-Brain Mapping of Direct Inputs to Midbrain Dopamine Neurons. *Neuron*, 74 (5):858–873, 2012.
19. Mateo Vélez-Fort, Charly V. Rousseau, Christian J. Niedworok, Ian R. Wickersham, Ede A. Rancz, Alexander P.Y. Brown, Molly Strom, and Troy W. Margrie. The stimulus selectivity and connectivity of layer six principal cells reveals cortical microcircuits underlying visual processing. *Neuron*, 83(6):1431–1443, 2014.
20. Sachie K. Ogawa, Jeremiah Y. Cohen, Dabin Hwang, Naoshige Uchida, and Mitsuko Watabe-Uchida. Organization of monosynaptic inputs to the serotonin and dopamine neuromodulatory systems. *Cell Reports*, 8(4):1105–1118, 2014.
21. Martin K. Schwarz, Annemarie Scherbarth, Rolf Sprengel, Johann Engelhardt, Patrick Theer, and Guenter Giese. Fluorescent-Protein Stabilization and High-Resolution Imaging of Cleared, Intact Mouse Brains. *PloS one*, 10(5):e0124650, 2015.
22. Suzana Herculano-Houzel, Bruno Mota, and Roberto Lent. Cellular scaling rules for rodent brains. *Proceedings of the National Academy of Sciences of the United States of America*, 103(32):12138–12143, 2006.
23. Yongsoo Kim, Kannan Umadevi Venkataraju, Kith Pradhan, Carolin Mende, Julian Taranda, Srinivas C. Turaga, Ignacio Arganda-Carreras, Lydia Ng, Michael J. Hawrylycz, Kathleen S. Rockland, H. Sebastian Seung, and Pavel Osten. Mapping social behavior-induced brain activation at cellular resolution in the mouse. *Cell Reports*, 10(2):292–305, 2015.
24. William Menegas, Joseph F Bergan, Sachie K Ogawa, Yoh Isogai, Kannan Umadevi Venkataraju, Pavel Osten, Naoshige Uchida, and Mitsuko Watabe-Uchida. Dopamine neurons projecting to the posterior striatum form an anatomically distinct subclass. *eLife*, 4: 1–30, 2015.
25. Casper Bo Gravesen Salinas, Tess Tsai Hsiu Lu, Sanaz Gabery, Kasper Marstal, Tomas Alanentalo, Aaron Jeffrey Mercer, Anda Cornea, Knut Conradsen, Jacob Hecksher-Sørensen, Anders Bjorholm Dahl, Lotte Bjerre Knudsen, and Anna Secher. Integrated Brain Atlas for Unbiased Mapping of Nervous System Effects Following Liraglutide Treatment. *Scientific Reports*, 8(1):1–12, 2018.
26. Daniel Furth, Thomas Vaissiere, Ourania Tzortzi, Yang Xuan, Antje Martin, Iakovos Lazaridis, Giada Spigalon, GIlberto Fisone, Raju Tomer, Karl Deisseroth, Marie Carlen, Courtney A Miller, Gavin Rumbaugh, and Konstantinos Meletis. An interactive framework for whole-brain maps at cellular resolution. *Nature Neuroscience*, 21:139–149, 2018.
27. Jin, Michelle and Nguyen, Joseph D and Weber, Sophia J and Mejias-Aponte, Carlos A and Madangopal, Rajtarun and Golden, Sam A. SMART: An open source extension of WholeBrain for iDISCO+ LSFM intact mouse brain registration and segmentation. *bioRxiv*, page 727529, 2019. doi: 10.1101/727529.
28. Asim Iqbal, Asfandyar Sheikh, and Theofanis Karayannis. DeNeRD: high-throughput detection of neurons for brain-wide analysis with deep learning. *Scientific Reports*, 9(1):1–13, 2019. ISSN 20452322.
29. Jun Ho Song, Woochul Choi, You Hyang Song, Jae Hyun Kim, Daun Jeong, Seung Hee Lee, and Se Bum Paik. Precise Mapping of Single Neurons by Calibrated 3D Reconstruction of Brain Slices Reveals Topographic Projection in Mouse Visual Cortex. *Cell Reports*, 31(8): 107682, 2020. ISSN 22111247. doi: 10.1016/j.celrep.2020.107682.
30. Nitin Agarwal, Xiangmin Xu, and M. Gopi. Automatic detection of histological artifacts in mouse brain slice images. In Henning Müller, B. Michael Kelm, Tal Arbel, Weidong Cai, M. Jorge Cardoso, Georg Langs, Bjoern Menze, Dimitris Metaxas, Albert Montillo, William M. Wells III, Shaoting Zhang, Albert C.S. Chung, Mark Jenkinson, and Annemie Ribbens, editors, *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*, pages 105–115, Cham, 2017. Springer International Publishing. ISBN 978-3-319-61188-4.
31. Michaël Barbier, Astrid Bottelbergs, Rony Nuydens, Andreas Ebneth, and Winnok H. De Vos. SliceMap: An algorithm for automated brain region annotation. *Bioinformatics*, 34(4): 718–720, 2018. ISSN 14602059. doi: 10.1093/bioinformatics/btx658.
32. Nicolas Renier, Eliza L. Adams, Christoph Kirst, Zhuhao Wu, Ricardo Azevedo, Johannes Kohl, Anita E. Autry, Lolahon Kadiri, Kannan Umadevi Venkataraju, Yu Zhou, Victoria X. Wang, Cheuk Y. Tang, Olav Olsen, Catherine Dulac, Pavel Osten, and Marc Tessier-Lavigne. Mapping of Brain Activity by Automated Volume Analysis of Immediate Early Genes. *Cell*, 165(7):1789–1802, 2016.
33. Maged Goubran, Christoph Leuze, Brian Hsueh, Markus Aswendt, Li Ye, Qiyuan Tian, Michelle Y Cheng, Ailey Crow, Gary K Steinberg, Jennifer A. McNab, Karl Deisseroth, and Michael Zeineh. Multimodal image registration and connectivity analysis for integration of connectomic data from microscopy to MRI. *Nature Communications*, 10(1):1–17, 2019. ISSN 20411723. doi: 10.1038/s41467-019-13374-0.
34. David M. Young, Clif Duhn, Michael Gilson, Mai Nojima, Deniz Yuruk, Aparna Kumar, Weimiao Yu, and Stephan J. Sanders. Whole-Brain Image Analysis and Anatomical Atlas 3D Generation Using MagellanMapper. *Current protocols in neuroscience*, 94(1):e104, 2020. ISSN 19348576. doi: 10.1002/cpns.104.
35. Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436 – 444, 2015.
36. Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. *Nature Methods*, 2019.
37. Drew Friedmann, Albert Pun, Eliza L. Adams, Jan H. Lui, Justus M. Kebschull, Sophie M. Grutzner, Caitlin Castagnola, Marc Tessier-Lavigne, and Liqun Luo. Mapping mesoscale axonal projections in the mouse brain using a 3d convolutional network. *Proceedings of the National Academy of Sciences*, 117(20):11068–11075, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1918465117.
38. Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
39. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 770–778, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90.
40. Alexander P. Y. Brown, Lee Cossell, Molly Strom, Adam L. Tyson, Mateo Vélez-Fort, and Troy W. Margrie. Analysis of segmentation ontology reveals the similarities and differences in connectivity onto L2/3 neurons in mouse V1. *Scientific Reports*, 11(4983), 2021.
41. Quanxin Wang, Song Lin Ding, Yang Li, Josh Royall, David Feng, Phil Lesnar, Nile Graddis, Maitham Naeemi, Benjamin Facer, Anh Ho, Tim Dolbeare, Brandon Blanchard, Nick Dee, Wayne Wakeman, Karla E. Hirokawa, Aaron Szafer, Susan M. Sunkin, Seung Wook Oh, Amy Bernard, John W. Phillips, Michael Hawrylycz, Christof Koch, Hongkui Zeng, Julie A. Harris, and Lydia Ng. The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas. *Cell*, 181(4):936–953.e20, 2020. ISSN 10974172.
42. Adam L. Tyson, Charly V. Rousseau, and Troy W. Margrie. brainreg: automated 3D brain registration with support for multiple species and atlases, August 2020.
43. Christian J. Niedworok, Alexander P.Y. Brown, M. Jorge Cardoso, Pavel Osten, Sebastien Ourselin, Marc Modat, and Troy W. Margrie. AMAP is a validated pipeline for registration and segmentation of high-resolution mouse brain data. *Nature Communications*, 7:1–9, 2016.
44. Mateo Vélez-Fort, Edward F. Bracy, Sepiedeh Keshavarzi, Charly V. Rousseau, Lee Cossell, Stephen C. Lenzi, Molly Strom, and Troy W. Margrie. A Circuit for Integration of Head- and Visual-Motion Signals in Layer 6 of Mouse Primary Visual Cortex. *Neuron*, 98(1):179–191, 2018. ISSN 08966273.
45. Helen Spiers, Harry Songhurst, Luke Nightingale, Joost de Folter, Roger Hutchings, Christopher J Peddie, Anne Weston, Amy Strange, Steve Hindmarsh, Chris Lintott, Lucy M Collinson, and Martin L Jones. Citizen science, cells and cnns – deep learning for automatic segmentation of the nuclear envelope in electron microscopy data, trained with volunteer segmentations. *bioRxiv*, 2020. doi: 10.1101/2020.07.28.223024.
46. Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, 2018. ISSN 15461726. doi: 10.1038/s41593-018-0209-y.
47. Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.
48. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.
49. Stephan Preibisch, Stephan Saalfeld, and Pavel Tomancak. Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics*, 25(11):1463–1465, 2009. ISSN 13674803.
50. Robert A.A. Campbell. BakingTray: Serial-section automated anatomy extension for ScanImage, 2020.
51. Robert A.A. Campbell, A. Blot, and Iguerard. StitchIt: Stitching of large tiled datasets, 2020.
52. François Chollet et al. Keras. `https://keras.io`, 2015.
53. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng.

TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

54. Nicholas Sofroniew, Talley Lambert, Kira Evans, Philip Winston, Juan Nunez-Iglesias, Grzegorz Bokota, Kevin Yamauchi, Ahmet Can Solak, ziyangczi, Genevieve Buckley, Matthias Bussonnier, Draga Doncila Pop, Tony Tung, Volker Hilsenstein, Hector, Jeremy Freeman, Peter Boone, alisterburt, Alan R Lowe, Christoph Gohlke, Loic Royer, Hagai Har-Gil, Mark Kittisopikul, Shannon Axelrod, kir0ul, Abhishek Patil, Abigail McGovern, Ariel Rokem, Bryant, and Heath Patterson. napari/napari: 0.4.4, January 2021.
55. Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv*, 2014.
56. Federico Claudi, Luigi Petrucco, Adam L. Tyson, Tiago Branco, Troy W. Margrie, and Ruben Portugues. Brainglobe atlas api: a common interface for neuroanatomical atlases. *Journal of Open Source Software*, 5(54):2668, 2020. doi: 10.21105/joss.02668.
57. Marc Modat, Gerard R. Ridgway, Zeike A. Taylor, Manja Lehmann, Josephine Barnes, David J. Hawkes, Nick C. Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine*, 98(3): 278–284, 2010.
58. S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache. Reconstructing a 3D structure from serial histological sections. *Image and Vision Computing*, 19(1-2):25–31, 2001.
59. Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.
60. Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Simon Hawkins, Sinhrks, Matthew Roeschke, Adam Klein, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, Marc Garcia, Jeremy Schendel, patrick, Andy Hayden, Daniel Saxton, Vytautas Jancauskas, Ali McMaster, Marco Gorelli, Pietro Battiston, Skipper Seabold, Kaiqi Dong, chris b1, h vetinari, and Stephan Hoyer. pandas-dev/pandas: Pandas 1.2.2, February 2021.
61. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
62. J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.
63. F. Claudi, A. L. Tyson, L. Petrucco, T.W. Margrie, R. Portugues, and T. Branco. Brainrender: a python-based software for visualizing anatomically registered data. *bioRxiv*, 2020. doi: 10.1101/2020.02.23.961748.

Materials and methods

All experiments were carried out in accordance with the UK Home Office regulations (Animal Welfare Act 2006) and approved by the establishments Animal Welfare and Ethical Review Board.

Sample preparation. All mice used were transgenic Cre-reporter (Ntsr1-Cre, GAD2-IRES-Cre, Rbp4-Cre & Penk-Cre) mice bred on a C57BL/6 background. The mice were anesthetized and an AAV Cre-dependent helper virus encoding both the envelope protein receptor and the rabies virus glycoprotein was stereotactically injected into visual cortex or retrosplenial cortex. Four days later, a glycoprotein deficient form of the rabies virus expressing mCherry was delivered into the same site. After ten further days, the animal was deeply anaesthetized and transcardially perfused with cold phosphate buffer (0.1 M) followed by 4% paraformaldehyde (PFA) in PB (0.1 M) and the brain left overnight in 4% PFA at 4 °C.

Imaging. All data was acquired using serial section two-photon tomography (8). To generate the data to pre-train the deep-learning model, fixed brains were embedded in 4% agar and placed under a two-photon microscope containing an integrated vibrating microtome and a motorized x-y-z stage as previously described (19). Coronal images were acquired via two optical pathways (red and blue) as a set of 6 by 9 tiles with a voxel size of 1 µm x 1 µm obtained every 5 µm using an Olympus 10x objective (NA = 0.6) mounted on a piezoelectric element (Physik Instrumente, Germany). Following acquisition, image tiles were corrected for uneven illumination by subtraction of an average image from each physical section. Tiles were then stitched using a custom FIJI (14) plugin (modified from (49)) and downsampled to 2 µm x 2 µm x 5 µm voxel size.

To generate the data to test the algorithm, data was acquired using a different, custom-built resonant-scanning system controlled by ScanImage (v5.6, Vidrio Technologies, USA) using BakingTray (50), a custom software wrapper for setting up the imaging parameters. Images were assembled using StitchIt (51). Both brains were imaged in a single acquisition using a Nikon 16x objective (NA = 0.8) , with a voxel size of 2.31 µm x 2.31 µm x 5 µm.

Cell candidate detection. To detect cell candidates (broadly defined as anything of sufficient brightness and of approximately the correct size to be a cell), initially data from the primary signal channel was processed in 2D (Fig. S1). Images were median filtered, and then a Laplacian of Gaussian filter was performed to enhance small, bright structures (e.g. cells). This filtered image was binarised using a threshold calculated for each image plane (mean of image plane + 10 x image plane standard deviation). The thresholded image was then passed to an ellipsoidal filter to remove noise. Every position of this spatial filter in which the majority (given by an input parameter, ellipsoidal filter overlap threshold fraction) of the filter overlaps with thresholded voxels was saved as a potential cell candidate. This is used to remove noise (from e.g. neurites).

All cell candidates that form continuous spatial structures were merged together, and classed as a single cell candidate. If the resulting cluster was too large to be a single cell (based on the input cell somata size parameter), then this cluster was split into individual cell candidates using an iterative ellipsoidal filter. Briefly, the ellipsoidal filter was applied to all voxels within the cell candidate cluster, and any resulting cell candidate coordinate positions were recorded. The thresholded image was eroded, and the filter was reapplied on the eroded set of candidate voxels. This process was repeated for ten iterations, or until there were no cell candidates remaining. This process ensures that densely labelled cells are split into individual cell candidates.

Once the final list of candidate cells is determined, the centroid of each cell candidate (based on the 3D mean coordinate of thresholded voxels) was calculated, and the coordinates were saved as an XML file. All of these steps were all carried out using the default software parameters (Table 5), with the exception of the 40 µm axial spacing dataset for which the axial extent of the ellipsoid filter was increased to 30 µm.

Table 5. Default cell detection parameters

Parameter	Value
In-plane cell somata diameter (⌀)	16 µm
Gaussian smoothing sigma	0.2⌀
Intensity threshold	Mean + 10 x SD
In-plane (lateral) ellipsoidal filter width	6 µm
Axial (z) ellipsoidal filter width	15 µm
Ellipsoidal filter overlap threshold fraction	0.6

Cell candidate classification using deep learning. Cell candidates were classified using a ResNet (39), implemented in Keras (52) for TensorFlow (53). 3D adaptations of all networks from the original paper are implemented in the software (i.e. 18, 34, 50, 101 and 152-layer) and can be chosen by the user, but the 50-layer network was used throughout this study. The general architecture of these networks is shown in Figs. S2 & S3.

To generate data to train the classification network, output from the candidate detection step (cell candidate coordinates) were manually classified using a custom FIJI (14) plugin, or an integrated tool (cellfinder-curate) using napari (54) that is supplied with the software. Expert annotators were presented with the raw data, with cell candidates represented by hollow spheres, centered on the cell candidate. By scrolling through the 2D planes of the 3D dataset, the experts could view the cell candidate in 3D before marking it as a cell or artefact. Candidates were determined to be cells or artefacts based on size, shape (including the presence of neurites) and the fluorescence level compared to the secondary autofluorescence channel (which was visualised at the same time). Candidates were labelled by three experts, labelling different brains, and a subset of labelled candidates were cross-checked between experts.

Image cuboids of 50 µm x 50 µm x 100 µm (resampled to 50 x 50 x 20 voxels) were extracted from both the primary signal, and secondary autofluorescence channels, centered on the coordinates of the manually classified cell candidate positions. To increase the size of the training set, data were randomly augmented. Each of the following transformations were applied with a 10% likelihood: (i) flipping in any of the three axes, (ii) rotation around any of the three axes (between 45° to 45°) and (iii) circular translation along any of the three axes (up to 5% of the axis length). The networks were trained using an NVIDIA TITAN RTX GPU with a batch size of 32 and the Adam (55) method was used to minimise the loss (categorical cross entropy), with a learning rate of 0.0001. Cell candidates were classified using the trained network, and saved as an XML file with a cell or artefact label.

Image registration and segmentation. To allow detected cells to be assigned an anatomical label, and for them to be analysed in a common coordinate framework, a reference atlas (Allen Mouse Brain Atlas, ARA, (41), provided by the BrainGlobe Atlas API (56)) was registered to the autofluorescence channel. This was carried out using brainreg (42), a Python port of the automatic mouse atlas propagation (aMAP) software (43), which itself relies on the medical image registration library, niftyreg (57). Firstly the sample brain was downsampled to the same voxel spacing as the atlas (10 µm isotropic) and was reoriented to the atlas template brain. These two images were then filtered to remove-high frequency noise (greyscale opening and flat-field correction). The images were firstly registered using an affine transform (*reg_aladin* (58)), followed by freeform non-linear registration (*reg_f3d* (57)). The resulting transformation was applied to the atlas brain region annotations (and a custom hemispheres atlas) to bring it into the same coordinate space as the sample brain.

Validation. To compare results of the algorithm to ground truth, two experts manually annotated each cell in two whole-brain images using the same cellfinder-curate tool used to generate the training data. The experts were shown the full-resolution images, with both channels displayed in different colors. The experts could scroll through the 3D images plane by plane, zoom in and out, and adjust contrast settings to best visualise cells in different brain areas. Experts annotated cells based on the same criteria as for generating the training data (shape, size and fluorescence intensity). Annotating a cell position displayed a hollow sphere on top of the image, around the cell coordinate. This was visible in multiple image planes to ensure that individual cells were not labelled more than once.

The cell classification network was retrained for the new datasets, and the algorithm was run with the default cell candidate detection parameters. The images were also registered to the ARA, and cell coordinates were assigned to brain regions for both the automated and manual cell counts. The different cell counting approaches were firstly compared by calculating the Pearson correlation coefficient using Pandas (59, 60). To assess the bias of the different approaches, they were compared by calculating the slope of the best fit line by fitting a linear model using scikit-learn (61).

Effect of varying axial sampling. To generate synthetic datasets with varying axial sampling, a single brain was downsampled in 3D by selecting every Nth image plane. For example, to generate a dataset with 20 µm sampling from the original dataset sampled at 5 µm, every fourth plane was used.

Visualisation. For visualisation of data in standard space, detected cells must be transformed to the atlas coordinate space. Firstly, the affine transform from the initial registration was inverted (using *reg_transform*). The sample brain was then registered non-linearly to the atlas (again using *reg_f3d*) and a deformation field (mapping points in the sample brain to the atlas) was generated (using *reg_transform*). This deformation field was applied to the coordinates of the detected cells for each sample, transforming them into atlas coordinate space.

Plots were generated using Matplotlib (62), and image visualisation was performed using napari (54) and brainrender (63).

Tyson, Rousseau, Niedwork *et al.* | bioRχiv | 9

Supplementary figures

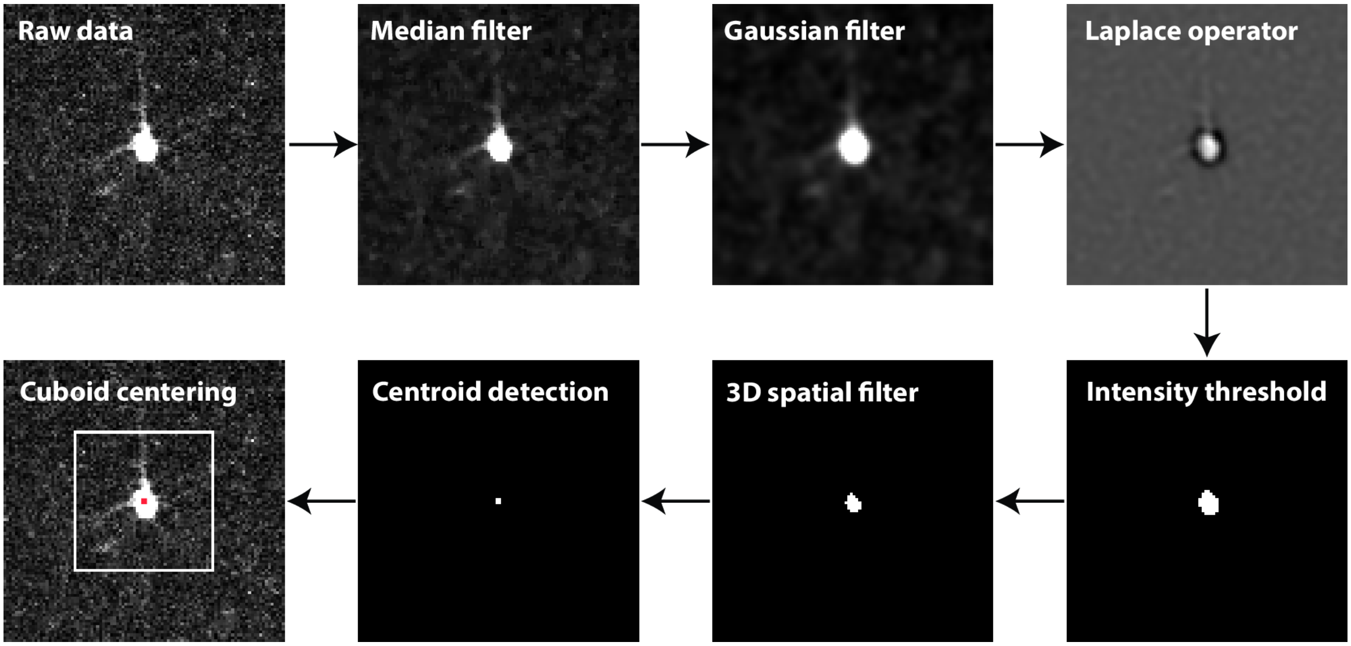

Fig. S1. Overview of the initial cell candidate detection steps, from raw data to a cuboid of data fed into the classification network. Upper row: from left to right, the raw image is median filtered to remove noise. A Laplacian of Gaussian is then performed to enhance small, bright structures such as the cell soma. Lower row: from right to left, the image is thresholded and a 3D ellipsoidal filter is used to remove small, non-cellular objects (not shown in this image plane). The centroid of the resulting object is then used to center the cuboid of data that it passed to the deep learning classification network. Images shown are $100\,\mu\mathrm{m}$ x $100\,\mu\mathrm{m}$, and the cuboid is $50\,\mu\mathrm{m}$ x $50\,\mu\mathrm{m}$ (and $100\,\mu\mathrm{m}$ in the third dimension).

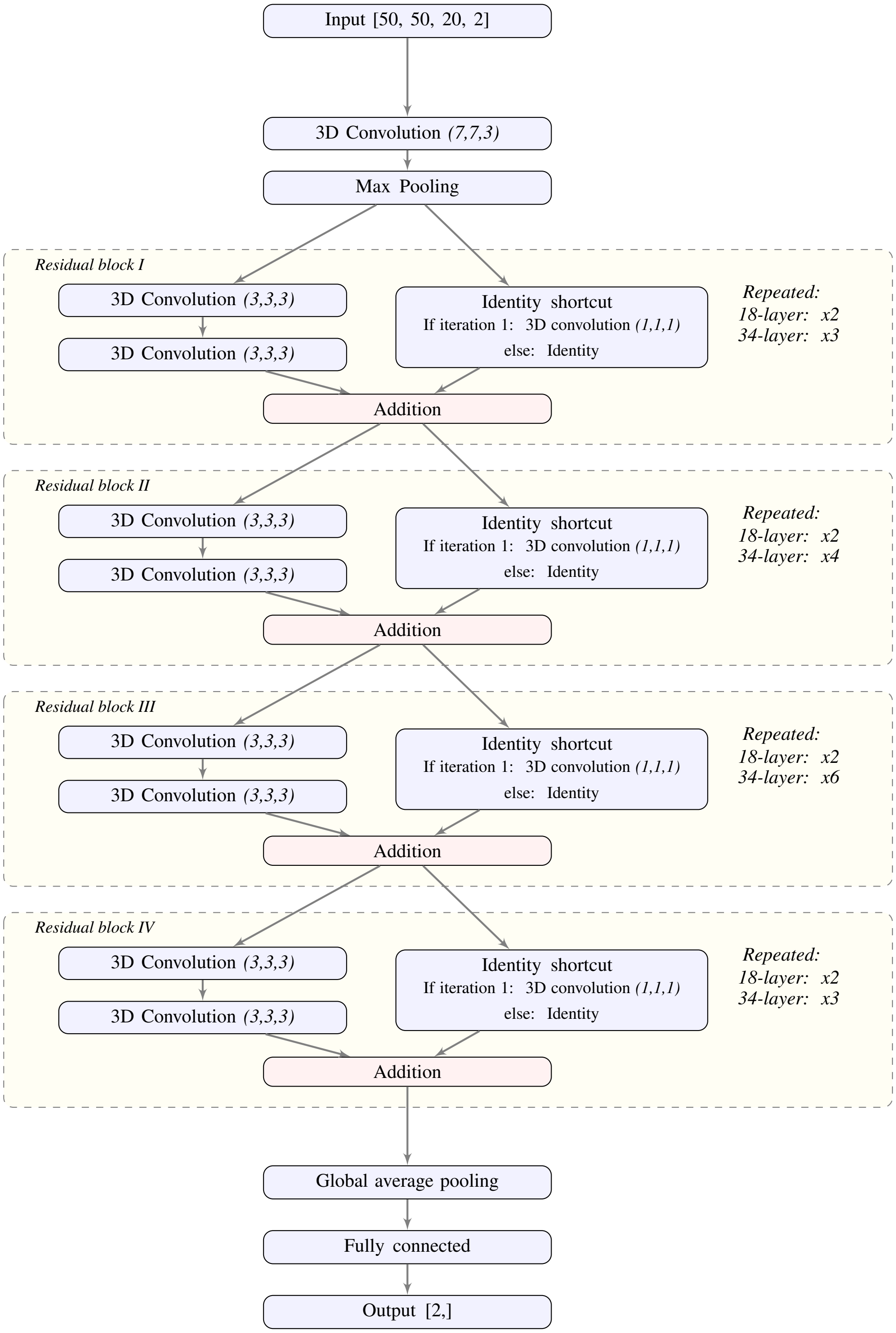

Fig. S2. Architecture of the 3D ResNet. 3D adaptation of the 2D networks from [38] which are available for use in the software.

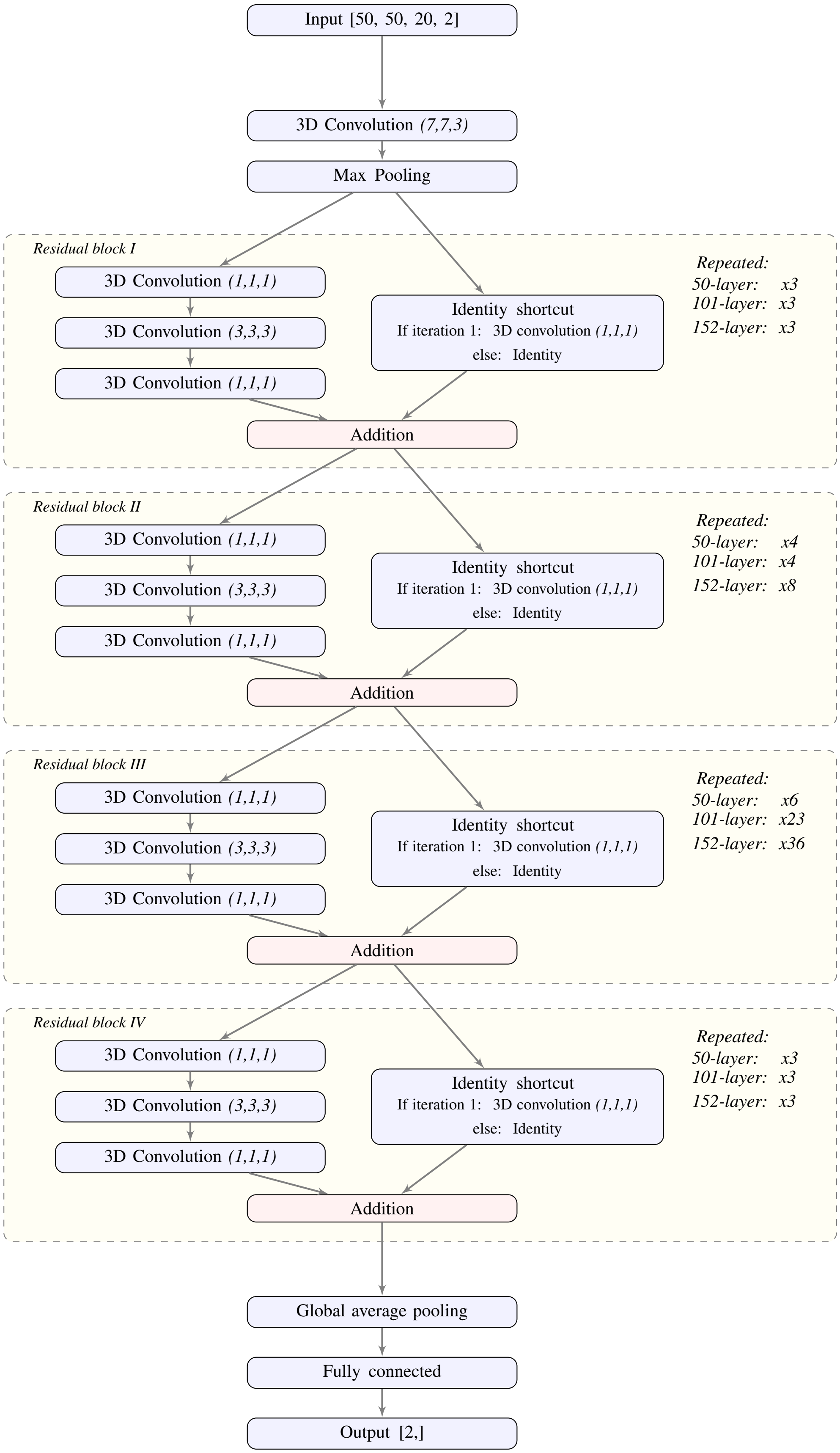

Fig. S3. Architecture of the bottleneck 3D ResNet. 3D adaptation of the bottleneck 2D networks from [38] which are available for use in the software. The 50-layer bottleneck network is used throughout this study, and is used for the pre-trained model supplied with the software.