scientific reports

OPEN

Utilizing 2D-region-based CNNs for automatic dendritic spine detection in 3D live cell imaging

Fabian W. Vogel[1], Sercan Alipek[1], Jens-Bastian Eppler[1], Pamela Osuna-Vargas[1], Jochen Triesch[1], Diane Bissen[2], Amparo Acker-Palmer[2], Simon Rumpel[3,4] & Matthias Kaschube[1,4]✉

Dendritic spines are considered a morphological proxy for excitatory synapses, rendering them a target of many different lines of research. Over recent years, it has become possible to simultaneously image large numbers of dendritic spines in 3D volumes of neural tissue. In contrast, currently no automated method for 3D spine detection exists that comes close to the detection performance reached by human experts. However, exploiting such datasets requires new tools for the fully automated detection and analysis of large numbers of spines. Here, we developed an efficient analysis pipeline to detect large numbers of dendritic spines in volumetric fluorescence imaging data acquired by two-photon imaging in vivo. The core of our pipeline is a deep convolutional neural network that was pretrained on a general-purpose image library and then optimized on the spine detection task. This transfer learning approach is data efficient while achieving a high detection precision. To train and validate the model we generated a labeled dataset using five human expert annotators to account for the variability in human spine detection. The pipeline enables fully automated dendritic spine detection reaching a performance slightly below that of the human experts. Our method for spine detection is fast, accurate and robust, and thus well suited for large-scale datasets with thousands of spines. The code is easily applicable to new datasets, achieving high detection performance, even without any retraining or adjustment of model parameters.

Neurons in the brain are connected via synapses to form intricate circuits, and the way information is processed and stored in these circuits depends strongly on the efficacy of synaptic connections. Most excitatory synapses in cortex are located on dendritic spines, which are small membranous protrusions from a neuron's dendrite. A dendritic spine typically receives input from a single excitatory synapse, and the size of the dendritic spine provides a proxy for synaptic efficacy[1–8]. Therefore, dendritic spines have been the focus of many lines of research, including studies on long-term potentiation (LTP) and long-term depression (LTD) of excitatory synaptic transmission[9,10] and on neurodegenerative diseases[11–14]. Advances in imaging technology enable the visualization of large populations of spines in vivo[4, 15–18], which, potentially, could provide us with an unprecedented detailed view on the simultaneous changes of thousands of synapses, for instance, during baseline conditions[19,20], or when an animal is involved in learning a task and forms a memory[21]. However, with the growth of such datasets, the prohibitive limitation is no longer the data acquisition, but rather the identification of spines in the recorded images. While hand-curated labeling is effective for small numbers of spines, such approach would be daunting for more recent large-scale recordings, thus requiring tools for the fully automated detection and analysis of large numbers of spines.

To extract spine positions, classical methods typically employed fixed, preselected image processing tools such as skeletonization[22], geodesic transforms[23], scale invariant feature transform (SIFT)[24] or generalized gradient vector flow (GGVF)[25]. Some studies extracted the dendritic shaft first, and then used this information to locate putative spines on it[26,27]. However, these methods are rule-based and are very sensitive to small variations in the underlying structure of the data (e.g. different shapes of spines, imaging noise, image sizes, resolution). Their

[1]Frankfurt Institute for Advanced Studies and Department of Computer Science and Mathematics, Goethe University Frankfurt, Ruth-Moufang-Straße 1, 60438 Frankfurt am Main, Germany. [2]Institute for Cell Biology and Neuroscience, Goethe University Frankfurt, Max-von-Laue-Str. 15, 60438 Frankfurt am Main, Germany. [3]Institute of Physiology, FTN, University Medical Center, Johannes Gutenberg University Mainz, Hanns-Dieter-Hüsch-Weg 19, 55128 Mainz, Germany. [4]These authors jointly supervised this work: Simon Rumpel and Matthias Kaschube. ✉email: kaschube@fias.uni-frankfurt.de

application to different datasets usually requires substantial amount of fine-tuning and expert knowledge, as such methods do not generalize well to new data obtained under different experimental conditions.

Deep convolutional neural networks (CNNs) are state-of-the-art in various fields of computer vision, including object recognition and medical image analysis. The ability of CNNs to detect complex patterns and to generalize well from specific training sets to novel data has been proven useful in a broad range of live-cell imaging applications, for instance detecting cancer[28,29] and immune cells[30], and for cell tracking in general[31]. CNNs have also been successfully applied to data at more macroscopic scales, e.g. for analyzing the motion[32] and pose[33,34] of individual animals and groups of animals. While some efforts have been made to use neural networks for dendritic spine detection[24,35–37], coming close to the robust detection performance reached by human experts has been a challenge.

Here, we utilize the framework of region proposal networks to devise a fully automated pipeline for dendritic spine detection in 3D volumes of brain imaging data of sparsely labeled neurons. The method leverages transfer learning by adopting a CNN that was previously trained on a general visual object recognition and localization task, and retraining this network with spine imaging data. The fact that the network is pretrained drastically reduces the amount of dendritic spine data necessary for training the model. Our pipeline first recognizes and localizes putative dendritic spines in 2D images. In case these images are part of a 3D imaged volume, this information is then integrated across the entire image stack to reveal the 3D locations of dendritic spines in that volume. To train and validate the model we generated a labeled dataset using five human expert annotators to account for the variability in human spine detection. We show that spine detection by our model is fast even for large numbers of spines and the detection performance achieved comes close to the performance reached by individual human experts. Moreover, without the need of further adjustment or training, spine detection generalizes well when applied to different datasets obtained in different laboratories using different labeling techniques and spatial resolutions. The code is publicly available[38]. Thus, by providing a robust, generalizable and scalable solution, our method addresses the existing bottleneck of large-scale automated spine detection.

Methods

Dendritic spine imaging dataset

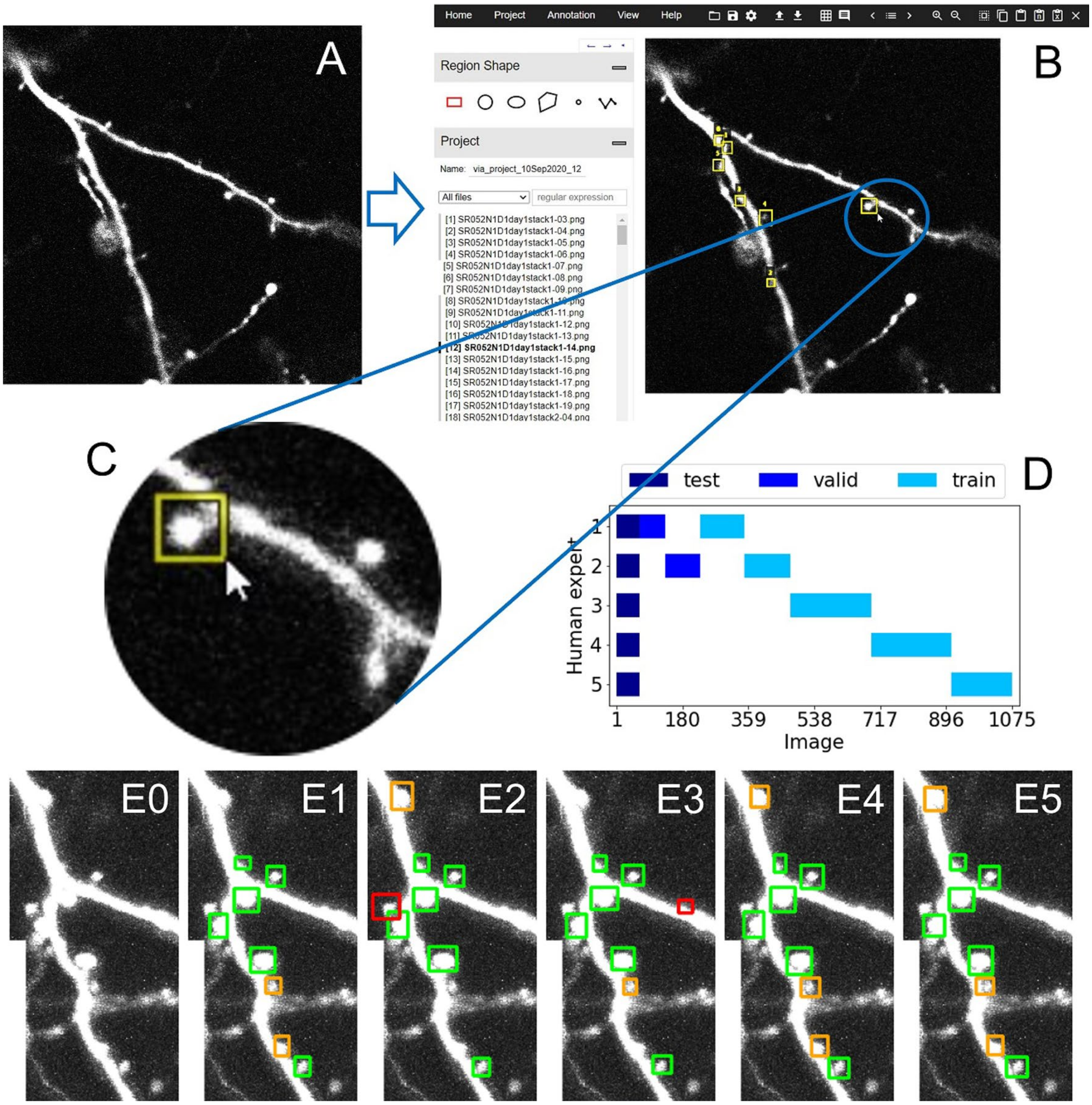
Our main dataset for developing and validating our analysis pipeline consisted of 55 two-photon image stacks recorded in the auditory cortex of a single mouse, containing dendritic branches of GFP-expressing neurons along with their dendritic spines (Fig. 1A). This dataset was published earlier[4,18] and details about data acquisition can be found there. The images were grayscale and had a size of 512×512 pixel corresponding to a linear extent of 51.2 μm. The distance in z-direction between two consecutive image slices of the same stack was 0.5 μm. To achieve a better distinction between the noisy background and the dendrites/spines, images were clipped at a maximum gray value of 255, which effectively resulted in gray value saturation in a smaller subset of pixels within dendrites and spine heads.

Optical slices (images) that did not contain dendritic spines were excluded from the stacks, leaving 1075 slices in total that contained at least one spine each. The number of slices varied across stacks, from 5 to 68 (mean ± SD: 19.5 ± 11.5). The 55 stacks were divided into a training dataset (for training the model), a validation set (for hyperparameter optimization, see below), and a test set (for performance assessment) containing 44, 6, and 5 stacks (non-overlapping) with a total of 844, 169, and 62 optical slices, respectively. According to the maximal vote of human annotations (see below), these sets contained 689, 116, and 55 dendritic spines, respectively. A tabular overview of our datasets is shown in Fig. 2. The training dataset provides the features that the network is learning in order to maximize the model performance on the validation dataset: While the network trains on the training samples, the hyperparameter choices are made with respect to an increasing validation performance. Finally, the test dataset was used to assess the model's performance on data not included in training and validation.

Manual labeling of data

The complete workflow of manual annotation is shown in Fig. 1. Five human expert annotators each labeled a subset of the 1075 optical slices semiautomatically using the VGG Image Annotator[39]. Each individual labeled between 228 and 282 images. A simple algorithm was used to set prelabels for putative spines in all slices, based on the labels for subsets of spines in single slices provided in[4] (Fig. 1B,C). A given dendritic spine in an image was marked by a rectangle, which the annotators were able to shift and resize to estimate the minimal bounding box containing this spine. For the majority of spines, manual labeling consisted of adjusting/correcting these prelabels. However, sometimes the insertion of new (and to a smaller degree also the elimination of existing) bounding boxes was necessary, as not all dendritic branches were labeled in the original study[4]. Note that it was necessary to include these spines and label them, as the presence of unlabeled true spines during training would confuse the network whose task is to learn to distinguish between 'spine' and 'no-spine'. Also note that in several cases the structure under consideration appeared difficult to interpret, even for expert annotators (see Fig. 1E0–5 for examples). To reduce such error sources, we used the 'four-eyes' principle to improve the quality of labeled data: All annotations were reviewed together with a second expert annotator and corrected, if necessary. This additional check increased the quality of our datasets and the detection performance of our human experts considerably.

The data subsets that were labeled by the five annotators were chosen to be mutually non-overlapping apart from a smaller subset that was labeled by all five annotators (Fig. 1D). This overlapping region allowed us to combine the detections of the different annotators to estimate a ground truth, which we then used to evaluate the detection performances of the individual annotators as well as of our model. Specifically, we created three different ground truths from these five different human annotations: A minimal, majority, and maximal ground



[1] SR052N1D1day1stack1-03.png
[2] SR052N1D1day1stack1-04.png
[3] SR052N1D1day1stack1-05.png
[4] SR052N1D1day1stack1-06.png
[5] SR052N1D1day1stack1-07.png
[6] SR052N1D1day1stack1-08.png
[7] SR052N1D1day1stack1-09.png
[8] SR052N1D1day1stack1-10.png
[9] SR052N1D1day1stack1-11.png
[10] SR052N1D1day1stack1-12.png
[11] SR052N1D1day1stack1-13.png
[12] SR052N1D1day1stack1-14.png
[13] SR052N1D1day1stack1-15.png
[14] SR052N1D1day1stack1-16.png
[15] SR052N1D1day1stack1-17.png
[16] SR052N1D1day1stack1-18.png
[17] SR052N1D1day1stack1-19.png
[18] SR052N1D1day1stack2-04.png

Figure 1. Generation of ground truth data by five human expert annotators based on prelabeling. (**A**) Example 2D raw data image showing two dendritic branches along with several dendritic spines. (**B**) Annotated spines from[4,18] used as prelabels for generating the ground truth data for the present study. (**C**) To achieve this, these prelabels are inserted into the labeling tool VIA[39], checked by our human expert annotators and adjusted further, where necessary (including deleting or creating new boxes). (**D**) Overview of the image sets labeled by the five human experts and their assignment to the training, validation, and test sets. Note that the test set was labeled by all five annotators to obtain a more objective ground truth. Training sets, in contrast, were non-overlapping to maximize the overall size of the training data. (**E0**–**5**): Example of an image section (**E0**) with the labels from all five experts (**E1**–**5**), illustrating the degree of consistency and variability across human experts. Green, labeling consistent across all experts; orange, the majority and red, the minority of experts.

Dataset	Training	Validation	Test	Total
Stacks	44	6	5	55
Image slices	844	169	62	1075
Dendritic Spines	689	116	55	860

Figure 2. Size of training, validation, and test dataset. Number of stacks, image slices, and dendritic spines for each of the three datasets, training, validation, and test.

truth consisting of spines detected by at least one, three, or five annotators, respectively, based on an overlap score of their bounding boxes defined below. Keeping the overlapping part relatively small allowed us to distribute the individual annotations over a larger fraction of the dataset.

Dendritic spine detection pipeline: overview

Our pipeline for automated spine detection in three-dimensional image stacks consists of several steps, shown in Fig. 3: (1) Converting stacks of images to the required format (Fig. 3A), (2) obtaining a bounding box for every identified spine along with a confidence value of detection by applying an adequately trained Faster R-CNN model[40] for each optical slice separately (Fig. 3B), and (3) integrating this information on detected spines across all slices in the 3D stack to further improve the robustness of detection, to interpolate the spine bounding boxes across the different slices, and to define the outline and location of each spine within the 3D stack (Fig. 3C). This outline, a 3D box, consists of a 3D volume confined in z by the range of optical slices a given spine was observed in (Fig. 3D) and for each optical slice confined in x and y by the bounding box. The fluorescent pixels inside this 3D volume can then be used to estimate properties of the synapse. For instance, the synaptic efficacy can be estimated by integrating the total fluorescence within this volume. In case the data consists of single image slices, step (3) is omitted.

To evaluate the performance of our pipeline, we compared its detection results with the hand-labeled ground truth data, described in the "Results" section.

For step (2) above, spine detection on image slices, we adopted a pretrained Faster R-CNN with feature extractor ResNeXt-101[41]. Its processing steps are shown in Fig. 4. The training of this network along with the choice of architecture and hyperparameter optimization is described in the next section "Network training".

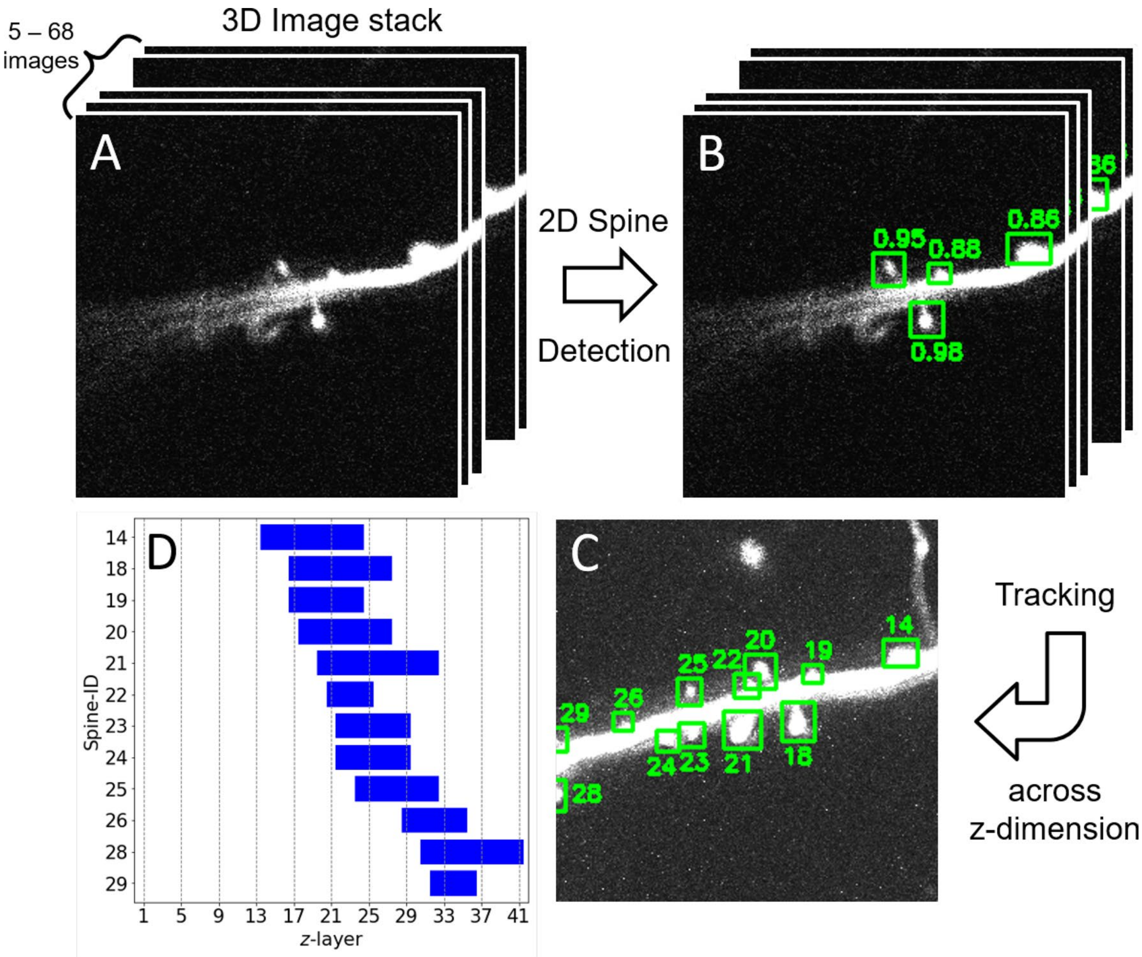


Figure 3. Overview of spine detection pipeline illustrated using one example image stack. (**A**) Input to the pipeline are 3D image stacks. In *z*-direction (depth) stacks vary from 5 to 68 image slices. (**B**) Spine detection is first carried out for each image slice, separately. For each spine a bounding box is estimated that contains the spine. Numbers indicate the estimated probability (confidence) that the corresponding box contains a spine. (**C**) Bounding boxes are then tracked across *z*-direction. Here, the numbers denote the IDs assigned to the different spines. Bounding boxes were averaged over the *z*-direction, as boxes varied slightly between the different *z*-slices. (**D**) The identified spines in 3D and their depth profiles. Each Spine-ID is equipped with the list of slices in *z* at which this spine was detected.

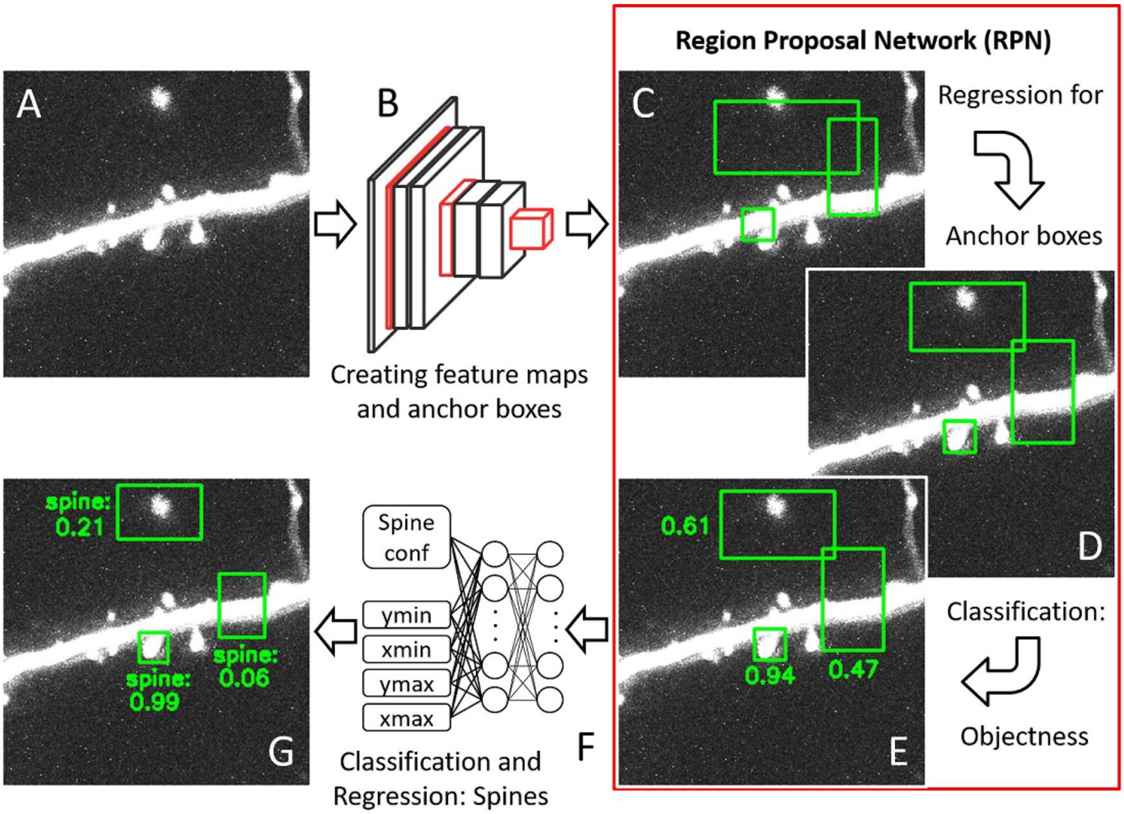


Figure 4. Detailed workflow of the 2D spine detection part (Fig. 3B) using a region proposal network (RPN). (**A**) A part of a raw image slice with several spines along a dendritic branch. (**B**) A sequence of multiple convolutional layers transform this image into sets of feature maps capturing relevant geometric properties in the image. (**C**) Anchor boxes of varying size and aspect ratio are generated (stride of eight, twelve anchor boxes per location). For the sake of clarity only three examples are shown. (**D**) The RPN then refines the location of these anchor boxes using regression. (**E**) For each box, the RPN calculates the probability (indicated by the numbers) that it contains an object (classification for objectness). (**F**) After region of interest (ROI) pooling, fully connected layers feed into (1) one output neuron representing the confidence of a spine being inside a given anchor box (classification for presence of spine) and (2) four output neurons estimating the top left and bottom right corner of this anchor box (via a regression step), (**G**) Final output image with bounding boxes around the detected spines along with confidence values (max, 1; min, 0). Note that the images used here have a lower than typical spine density to simplify the illustration.

Network training

Testing a variety of neural network architectures combined with different feature extractors, we found that using the Faster R-CNN[40] with feature extractor ResNeXt-101[41] as our base network provided promising results and, thus, we focus on this architecture in the following. The model was pretrained on the MS COCO dataset[42] for visual object recognition and afterwards trained on our spine dataset. Pretraining established the model's general ability to detect a broad range of visual objects, along with the necessary sensitivities for visual features on various spatial scales and levels of complexity. Adapting such a pretrained model enabled us to cope with a relatively small amount of training data (and training time), while achieving a high detection performance on tests with novel spine imaging data.

The network consists of two main parts: First, the Region Proposal Network (RPN), which generates an array of candidate bounding boxes (N = 50,700), the so called anchors (Fig. 4C). Anchor positions are further refined by regression and for each anchor a confidence score is computed expressing whether it contains an object or not (Fig. 4D, E). Applying non-maximum suppression (using the intersection over union (IoU) score with threshold 0.7) and based on the highest confidence scores, a subset of at most $N^{\mathrm{RPN}} = 1000$ of these boxes, the so called region proposals, are selected and fed into the second part of the network, the Box Classifier (BoxC) (Fig. 4F). This part performs Region of Interest (RoI) Pooling, refinement, and classification through a set of fully connected layers to obtain a reduced and final list of $N^{\mathrm{BoxC}} = 80$ bounding boxes with their confidence of containing a dendritic spine (Fig. 4G). Both networks were retrained jointly.

Training the first part, the RPN, involves a comparison between all the region proposals (i.e. the potential anchor boxes) and the ground truth boxes given by the labeled data. For each ground truth box the region proposal with the highest IoU overlap gets assigned the label 1, if this overlap exceeds a threshold $IoU \geq 0.7$. The remaining boxes with an IoU overlap below 0.3 for all ground truth boxes are not sufficiently close to any of the ground truth boxes, and they receive the label 0. All other boxes, which are assigned to neither of these labels, do not contribute to the training objective; thus, are not included in the loss calculation. This procedure is in line with the general training scheme used for Faster R-CNN[40]. The loss function then comprises two contributions: First, the objectness loss $\mathscr{L}_{obj}$ penalizing if the labels (object vs. non-object) of the anchors are correct, and second, the regression loss $\mathscr{L}_{regr}$ that penalizes deviations in the spatial position and size of anchor boxes. Both losses are normalized by the corresponding number of proposal regions. Their weighted sum yields the overall loss $\mathscr{L}^{RPN}$ ($\mathscr{L}_{regr}$ is weighted by a factor $\alpha = 2$ relative to $\mathscr{L}_{obj}$ to prioritize regression over classification).

Training the second part involves training the box classifier, whose structure is similar to the structure of the RPN, but placed at a different location within the network. Fig. 4C–E shows the location of the RPN and Fig. 4F that of the box classifier. The loss of the Box Classifier is denoted with $\mathscr{L}^{\mathrm{BoxC}}$ and is calculated similar to the loss $\mathscr{L}^{\mathrm{RPN}}$ of the RPN. However, the mapping between predictions and ground truths differs slightly as the set of predictions of the Box Classifier consists of only $N^{\mathrm{BoxC}} = 80$ refined boxes, as opposed to the $N^{\mathrm{RPN}} = 1000$ anchor boxes considered for the RPN loss, above, requiring an adjustment of the normalization constants. All other parts of the $\mathscr{L}^{\mathrm{BoxC}}$ calculation are analogous to that of $\mathscr{L}^{\mathrm{RPN}}$.

Finally, to avoid large weights inside the neural network, an additional L2 regularization loss $\mathscr{L}^{\mathrm{regu}}$ is added with a factor of $\lambda = 3 \times 10^{-6}$. The combined loss function then reads:

$$\mathscr{L} = \mathscr{L}^{RPN} + \mathscr{L}^{BoxC} + \lambda \mathscr{L}^{regu} \tag{1}$$

We trained the model using a batch size of 1 and gradient descent with momentum of 0.6 and learning rate of $\gamma = 1 \times 10^{-3}$. By visual inspection of the approximate spine sizes in the labeled images we chose the following parameters for the anchor generator: Scales of 8, strides of 4, 8, 16, 32, and 64 pixel, and aspect ratios of 0.5, 1.0, and 2.0. We used in-place data augmentation to increase the complexity of the dataset without increasing its size. To this end, we randomly applied a right-turning 90 degree rotation, vertical and horizontal flips, leading to eight possible different orientations of the original image, followed by adding Gaussian white noise and Gaussian blur, each with probability 0.5. In total, the network was trained for 14,348 iterations (17 epochs). To find optimal values for the hyperparameters momentum, learning rate, weight decay and the in-place data augmentation, we conducted a two-staged grid search, starting with a coarse grid followed by a finer sampling within the hyperparameter space.

All computations related to CNNs were conducted in Pytorch[43]. Training for 17 epochs took slightly less than one hour on an Nvidia RTX A6000 GPU. Our model took on average 0.053 s to make predictions on one optical slice. As it will always return 80 bounding boxes for spines, no matter how low the confidence score is, the detection time does not depend on the number of visible spines. Tracking one detection into the next frame took 0.01 s, which is negligible in comparison to the detection time. In all images of our dataset the number of spines never exceeded 26. In the case that more than 80 spines are visible, the image can be split up into n parts so that each part only contains less than 80 spines. This will multiply the detection time by a factor of n. Alternatively, this upper bound can be manually increased (requiring a retraining of the model).

Evaluation metric 2D

The evaluation of the accuracy of spine detection is based on the overlap between the detected and the ground truth boxes. To compute this overlap we defined the Intersection over Minimum, IoM_{xy}, for image slice i in the xy-plane as

$$IoM_{xy}(D_i, G_i) = \frac{\text{Area of } D_i \cap G_i}{\min\{\text{Area of } D_i, \text{Area of } G_i\}} \in [0, 1]\,, \tag{2}$$

where D_i is a detection box and G_i a ground truth box in this image slice. We found that this metric achieves more accurate results compared to the more commonly used metric Intersection over Union (IoU). This appears to be due to the fact that dendritic spines are small compared to the entire image so slight positional differences in the estimated bounding boxes can significantly affect the value of the IoU. Note that this issue is not solved simply by reducing the detection threshold of the IoU, as this produces more false positives. Figure 5A–C and the schematic Eq. (6) illustrate the difference between the IoU and the IoM_{xy}. Figure 5D–H illustrates the IoU and IoM_{xy} scores using hand-crafted bounding boxes that were chosen to emphasize the differences between these measures. In our data we found that using the IoM_{xy} typically yielded more robust results. In few cases the IoM_{xy} can be misleading if the bounding box is considerably larger than the ground truth. Thus, we introduced an area threshold to remove all bounding boxes exceeding a specific size of 2000 pixel (20 μm^2). This value is chosen according to the observed statistical spine properties: Most spines had an area between 100 and 700 pixel (1–7 μm^2), the largest spine in our labeled dataset reached an area of 1300 pixel (13 μm^2). As our model is trained to minimize the loss that depends on the exact position of the bounding box, such cases are extremely rare. We thus conclude that the metric IoM_{xy} is better suited for the purpose of spine detection as it represents the quality of human annotations and detections more accurately.

Integration across imaging depth

When applied to 2D data (single images), our method estimates dendritic spines as described up to this point. When a 3D image stack is provided, our tool integrates the information across depth, as described in the following.

If an image stack is provided, we use a tracking algorithm to combine detections across layers within the stack. This improves the robustness of detection (in the case a misdetection occurred in a few layers), and allows us to extract the x-, y- and z-coordinates of identified spines.

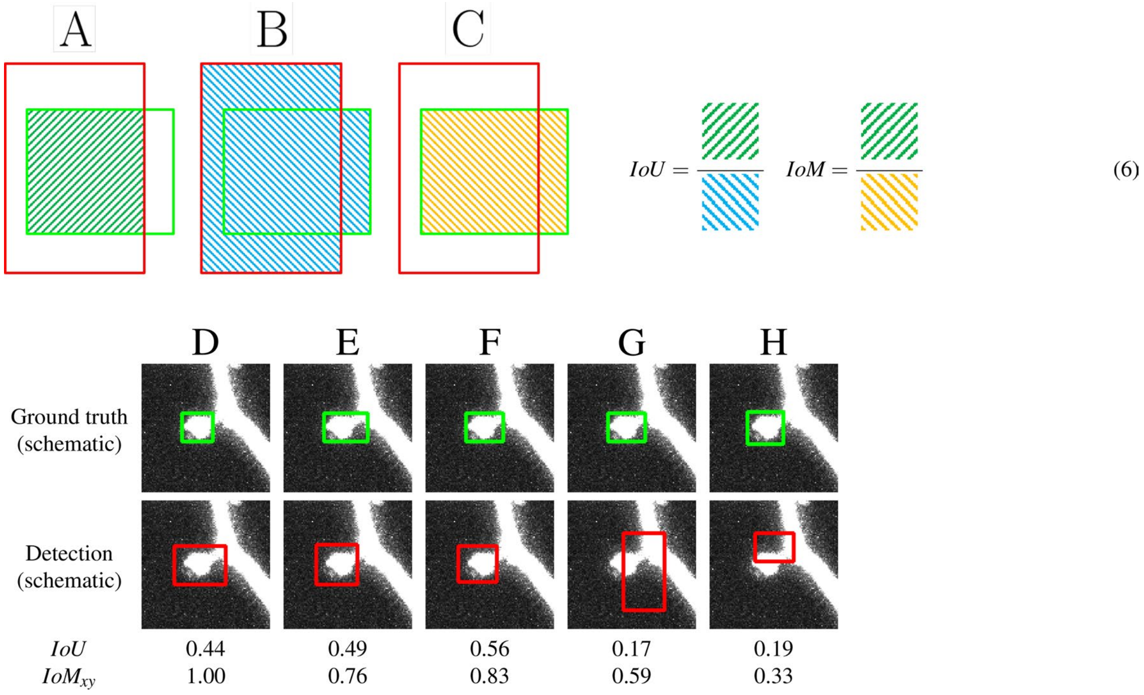


Figure 5. Intersection over Minimum (IoM) is better suited than Intersection over Union (IoU) for evaluating spine detection accuracy. (**A**)–(**C**) Given a ground truth box (green) and a detection box (red) their overlap can be defined by the *IoU* or the *IoM*, both of which are defined in the schematic equation (right, Eq. (6)), where the squared symbols colored in green, blue, and yellow denote the areas of intersection (**A**), union (**B**) and minimum area (**C**), respectively. Note that the two equations differ only in their denominator. (**D**)–(**H**): Illustration of the difference between *IoU* and IoM_{xy}. For fairly accurate detections (**D**–**F**), the *IoU* can be quite low, while the IoM_{xy} appears to reflect better the detection accuracy. For seemingly false detections (**G**, **H**) the IoM_{xy} is typically low (**H**), but can be inaccurate in rare cases (**G**). Note that the bounding boxes depicted here are drawn manually to illustrate the difference between *IoM* and *IoU*.

The algorithm proceeds as follows. First, the model yields a list of bounding boxes, each equipped with a confidence score (Fig. 4G). Next, the number of model predictions is reduced to only keep the relevant detections: Only boxes that have a confidence greater than the commonly used prediction probability threshold of 0.5 are considered real detections. If multiple detections with a pairwise IoM_{xy} of more than 0.5 exist, all those detections except the one with the highest detection probability will be removed.

In the last step, we devised a simple tracking algorithm that converts this set of remaining detections from the different optical slices into a list of identified spines for that stack. It starts with iterating over the *z*-axis from top to bottom. In each iteration, the identified spines from the current optical slice are compared to those in the next slice, using the IoM_{xy}-score. Pairs of identified spines with the highest IoM_{xy}-scores are considered to belong to the same spine if this score exceeds the threshold of 0.5. After connecting all spines, some spines detected in the next slice may be left without any assignment, in which case this spine was registered as a new spine. To account for possible missed detections in individual slices, the tracking of a given spine only stopped after two consecutive missing detections.

Evaluation metric 3D

Once spines have been tracked across depth in an image stack, we can define a bounding box in 3D along with a measure to compare such 3D boxes between model prediction and ground truth and between annotators and ground truth.

Each 3D detection box *D* consists of multiple boxes D_i which correspond to the box in image slice *i*. The values of *i* are taken from the interval $[z_1^D, z_2^D]$ representing the boundaries of the detection *D* in the *z*-dimension. Analogously, each ground truth box *G* is assembled by 2D boxes G_i for $i \in [z_1^G, z_2^G]$. The intervals are returned by the tracking algorithm above (see also Fig. 3D). Analogously to the IoM_{xy}, we defined the IoM_z representing the percentage overlap in *z*-direction of these intervals:

$$IoM_z(D, G) = \frac{\max\{0, \min\{z_2^D, z_2^G\} - \max\{z_1^D, z_1^G\}\}}{\min\{z_2^D - z_1^D, z_2^G - z_1^G\}} \in [0, 1] \quad (3)$$

Further, we computed the IoM_{xy} for 3D detection and ground truth boxes *D*, *G* by averaging in *z*-direction over all associated 2D boxes:

$$IoM_{xy}(D,G) = IoM_{xy}\left(\frac{1}{z_2^D - z_1^D}\sum_{i=z_1^D}^{z_2^D} D_i, \frac{1}{z_2^G - z_1^G}\sum_{i=z_1^G}^{z_2^G} G_i\right)$$

These two scores, IoM_{xy} and IoM_z, were then combined using the $F_{0.5}$-score. The parameter $\beta = 0.5$ is chosen to increase the importance of IoM_{xy} relative to IoM_z. The final IoM-score given a 3D detection D and a 3D ground truth label G is then defined as:

$$IoM(D,G) = F_{0.5}(IoM_{xy}(D,G), IoM_z(D,G)) = \frac{5 \cdot IoM_{xy}(D,G) \cdot IoM_z(D,G)}{IoM_{xy}(D,G) + 4 \cdot IoM_z(D,G)} \in [0,1] \tag{4}$$

We defined true positives (TP), false positives (FP) and false negatives (FN) similar to the common definition but using the IoM instead of the IoU to decide if a detection is correct or not. The specific definitions are as follows:

$$\begin{aligned} \text{True positive } \mathrm{TP}^{3D} &:= \text{Detection } D\text{, such that } \exists G : IoM(D,G) \geq 0.5 \\ \text{False positive } \mathrm{FP}^{3D} &:= \text{Detection } D\text{, such that } \forall G : IoM(D,G) < 0.5 \\ \text{False negative } \mathrm{FN}^{3D} &:= \text{Ground truth } G\text{, such that } \forall D : IoM(D,G) < 0.5 \end{aligned}$$

Finally, to evaluate the overall detection performance, we defined the $\boldsymbol{F_1}^{\mathbf{3D}}$**-score**, which is similar to the standard F_1-score, but with the above definitions of TP^{3D}, FP^{3D} and FN^{3D}. Given a list of detections $\mathscr{D} = \{D^{(j)}\}$ and a list of ground truth labels $\mathscr{G} = \{G^{(k)}\}$, this score is given by

$$F_1^{3D}(\mathscr{D}, \mathscr{G}) = \frac{2 \cdot \mathrm{TP}^{3D}}{2 \cdot \mathrm{TP}^{3D} + \mathrm{FP}^{3D} + \mathrm{FN}^{3D}}. \tag{5}$$

The F_1^{3D}-score was used both for comparing our model in 3D against the ground truth and for comparing the detections of individual annotators against the ground truth.

Results

Spine labels varied slightly across the five expert annotators (Fig. 1E1–E5), suggesting that for some cases it was difficult to decide between spine or no spine, even for a trained human expert. To account for this ambiguity we considered three alternative definitions of 'ground truth' data, against which we evaluated the performance of both our individual experts and our model. The 'minimal ground truth' comprised all spines that were identified by at least one of the experts, while the 'majority ground truth' comprised all spines identified by at least three experts, and 'maximal ground truth' the set of spines all five experts agreed on. Our assessment of performance focused mostly on the majority ground truth, as we expect it to reflect the most accurate estimate of the true set of spines. The number of TP, FP and FN for each human expert when compared to the majority ground truth are displayed in the table in Fig. 6.

Before investigating the overall performance of our model on the three-dimensional data stacks, we first examined the accuracy of spine detection when applying the model to individual two-dimensional image slices. After hyperparameter optimization the model's performance on the validation dataset, based on the standard F_1-score and averaged over four runs, reached 0.802 ± 0.012 (mean ± SEM). In comparison, the performance on the test dataset was slightly lower for all three types of ground truths: 0.747 ± 0.032 (minimal), 0.758 ± 0.028 (majority) and 0.741 ± 0.025 (maximal). A gap between validation and test performance is expected given the relative small number of image stacks within these two datasets and may in parts also reflect subtle differences across stacks regarding biological properties or the experimental conditions.

To leverage the full potential of the volumetric spine imaging datasets, we added a simple tracking scheme across the layers of the stack to integrate the detection information across the different image slices within the stack. This strategy allowed us to compensate for some of the detection failures or inaccuracies in the individual slices, resulting in overall more robust spine detection performances. To quantify this performance, we defined a novel three-dimensional performance metric, the F_1^{3D}-score (see "Methods"), and applied it to detections made on the test data (which was neither used for training nor hyperparameter optimization). Figure 7 summarizes the results for the detection performance of our model in three dimensions for the three types of ground truth, and

Human expert	1	2	3	4	5
TP	59	57	56	60	60
FP	2	2	2	4	1
FN	1	3	4	0	0

Figure 6. Degree of consistency in spine labeling across human experts. Number of true positives (TP), false positives (FP) and false negatives (FN) on the test dataset compared to the majority ground truth for each of the five human experts. Note that the number of FP will be zero and the number of FN significantly higher when compared to the maximal ground truth, whereas the reverse happens when compared to the minimal ground truth.

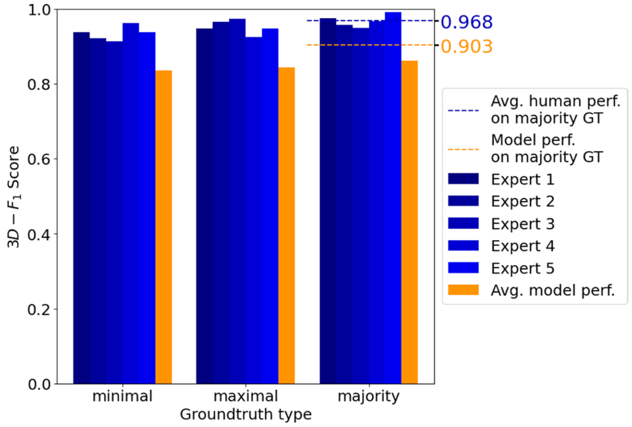


Figure 7. Automated spine detection reaches a performance slightly below human-level. Detection performance ($F_1^{3D}(\mathscr{D}, \mathscr{G})$, see "Evaluation metric 3D" section) of our model compared to the performance achieved by five human experts. To assess performance, detection is compared with three different types of ground truth (GT): minimal (left), maximal (middle) and majority (right), defined by the set of spines detected by at least one, all five and at least three experts, respectively. The model performance is averaged over four randomly initialized training runs. Using the hyperparameters with the best two-dimensional F_1-score on the validation set yields a model performance of 0.836 ± 0.025 (minimal), 0.843 ± 0.0082 (maximal) and 0.862 ± 0.0087 (majority ground truth) (orange bars). Even slightly higher test results were achieved for longer training times (26 vs. 17 epochs; orange dashed line, 0.903).

in comparison to the performance achieved by the five human expert annotators. For the model, the F_1^{3D}-scores for the three types of ground truths, averaged over four randomly initialized training runs, were 0.836 ± 0.025 (minimal), 0.843 ± 0.0082 (maximal) and 0.862 ± 0.0087 (majority). Note that these performances were achieved using the hyperparameters for which we obtained the maximal F_1-score for the two-dimensional detection in the image slices of the validation dataset. We noticed, however, that when training the model longer, this typically resulted in even higher F_1^{3D}-scores on the test dataset. For instance, for the majority ground truth, our model achieved a F_1^{3D}-score of 0.903 when trained over 26 instead of 17 epochs (orange dashed line in Fig. 7).

In comparison to the model, the human annotators reached slightly higher F_1^{3D}-scores with an average value of 0.968 (blue dashed line in Fig. 7). Figure 8 depicts examples of correctly detected spines along with a few examples of misdetections by the model. Examples of the latter include, for instance, cases of close proximity, which the model counts as a single spine due to the high overlap of two bounding boxes. Additionally, we sometimes observed less accurate detections for small spines that are located directly above or below a dendrite and only have a tiny expansion in the x-y plane. However, once trained, the time it took the model to process the data was substantially shorter compared with human experts. While our well-trained human annotators reached rates of 12 s per frame (for frames containing only few spines), the model processed a frame in 0.053 s (independent of spine number) using an Nvidia RTX A6000. This fast automated detection of spines makes the model readily applicable to large datasets.

Next, to test the model's robustness and generalizability, we applied it (without any further adjustment) to novel datasets. First, the model was applied to image stacks obtained from a different mouse (same experimental setup) that was neither used for training nor validation of the model. The example shown in Fig. 9A illustrates the overall high accuracy of the model, i.e. its detection of spines and their delineation with bounding boxes, and highlights the robustness of the model against animal-to-animal variability, noise and various background structures. Notably, the model was neither confused by axons (marked in red in Fig. 9A for display purposes) nor by other high fluorescence structures of similar size (e.g. the many puncta and protrusion-like structures in Fig. 9A). Validating the model detections based on spine annotations in maximum intensity projections by a human expert yielded an F_1-score of 0.969. These results suggest that the detection method generalizes well to novel datasets from different specimen obtained under similar experimental conditions.

Second, as a more challenging test, in Fig. 9B we applied the model (again without any further adjustment or training) to an entirely different spine imaging dataset that was obtained by a different laboratory in a different system (organotypic slice cultures from hippocampus vs. auditory cortex in vivo) using a different GFP label and a different microscope system with a different z-resolution (0.66 µm vs. 0.5 µm). As can be seen in Fig. 9B almost all spines were identified correctly by our model, achieving an F_1-score of 0.815 on this completely new dataset. Most of the missed spines were close to another detected spine in the *xy*-plane, and could potentially be distinguished when processing different *z*-layers. This suggests that our model generalizes well and can detect spines with high performance even in entirely novel datasets.

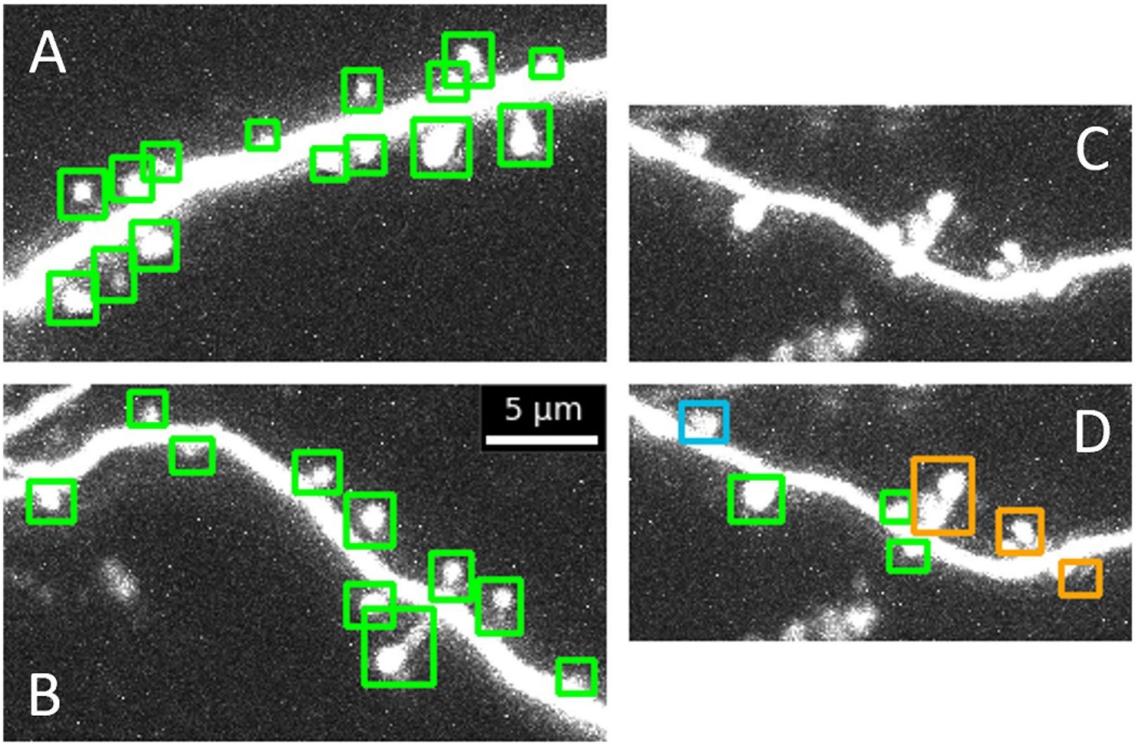


Figure 8. Examples of correctly identified spines and detection errors in novel data (from a mouse that was not used for training, validation or testing of the model). Images and bounding boxes are shown as maximum intensity projections (MIP) (which for the bounding boxes is basically equivalent to taking a 2D slice through the 3D box identified). (**A**), (**B**) Examples of correctly detected spines, (**C**) An example of an image section in which spine detection is particularly challenging. (**D**) Model results: some spines are correctly detected (green), but one is missed (blue) and several nearby spines are merged into a single detection (orange). Note that examples like these are rare in our data, but shown here to illustrate potential limitations of our model.

Discussion

We developed an efficient analysis pipeline to detect large numbers of dendritic spines—a proxy for excitatory synaptic connections—in fluorescence imaging data. The core of our pipeline is a deep convolutional neural network, which is pretrained on the task of object detection, and which we adjusted and trained specifically to detect spines in 2D images of sparsely fluorescent-labeled neurons. In case the data consists of 3D image stacks the detection algorithm is combined with an algorithm for tracking the identified spines across depth, which further improves the overall robustness of spine identification and enables us to reconstruct the outlines and positions of spines in 3D volumes of brain tissue. To train the model we used labeled data curated by five independent human expert annotators and we combined the overlapping parts of these five sets of annotations to estimate ground truth labeled data, which we used to test the detection performance of both our model and the individual annotators. For evaluation in 3D we devised the F_1^{3D}-score, an extension of the widely used F_1-score, well suited for comparing the 3D bounding boxes of detected spines with those in the ground truth data. We found that our method is fast and accurate, achieving a robust performance on the ground truth test data that is close to human-level performance. Most importantly, our method achieved accurate results also when applied to novel datasets that were obtained by a different laboratory in a different neural tissue, and without the need for any adjustment or retraining of the model, suggesting that it could be applicable with similar performance to a broad range of spine imaging data. The code for our method is available on GitHub[38].

When generating labeled data for training our network, we deliberately chose the labeled sets to be largely non-overlapping among our five annotators, while overlap was enforced in a subset of the data, allowing us to accurately estimate ground truth for testing and performance evaluation using that subset. This strategy was chosen in order to efficiently use the available resources for labeling. For network training it was advantageous to keep the other labeled parts non-overlapping, as this allowed the network during training to sample from a much broader set of different spines than it would have been possible if all labeled sets were overlapping among the five annotators while using the same total number of annotations. Similarly, also the validation dataset, on which the model hyperparameters were optimized, consisted of non-overlapping labeled data. While it is expected that the labeling in these non-overlapping parts of the datasets are more prone to noise and personal biases in the annotation procedure, this seems more than compensated by the larger number and diversity of labeled spines. Moreover, incorporating such biases during training may very well improve the overall robustness and generalizability of our spine detection method, which is particularly useful when applying it to novel types of datasets.

Several previous studies have used neural networks for automated spine detection[24,35,36]. The method by Smirnov et al.[36] first extracts features and then trains a neural network on these features to detect spines, instead

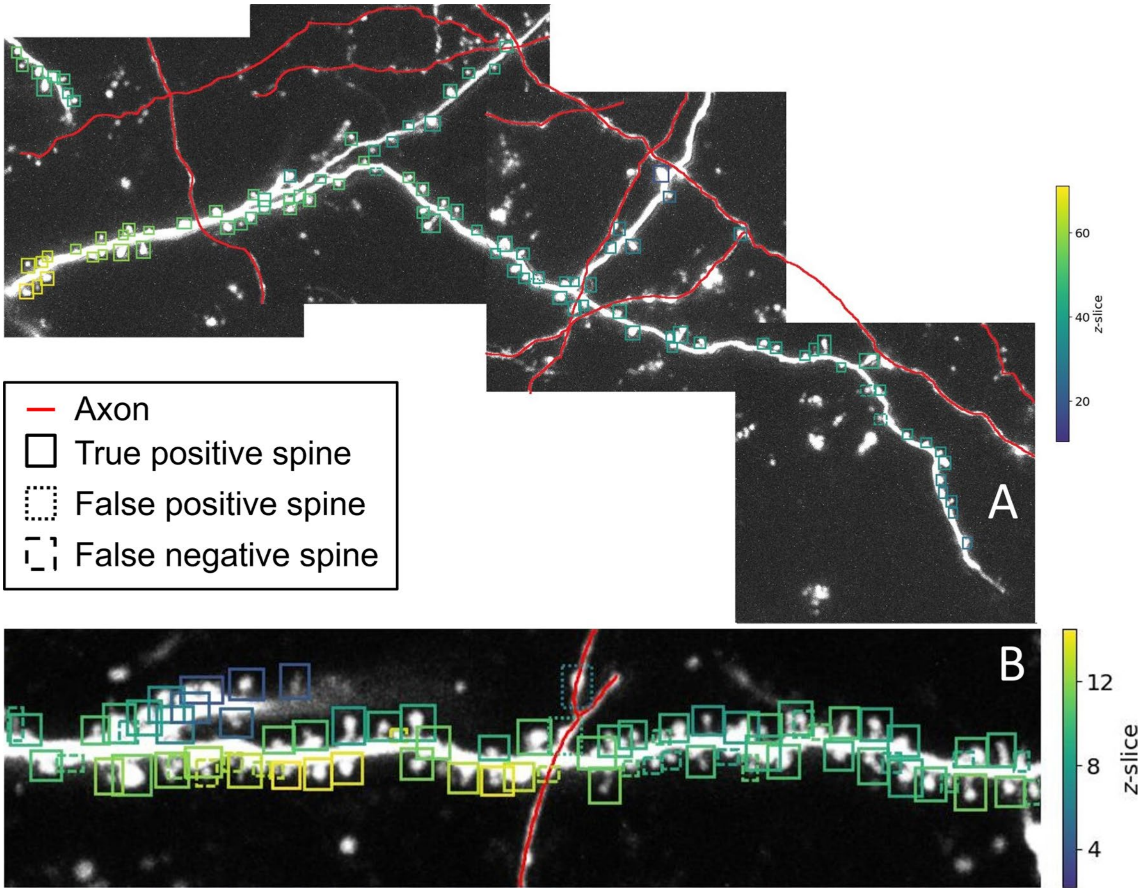


Figure 9. Large-scale detection of spines in 3D and its generalization to different datasets. (**A**) Application of detection pipeline to a new dataset from another animal that was not used for training, validation or testing of the model (same experiments as in previous figures, i.e. in vivo two-photon imaging in mouse auditory cortex; laboratory of S. Rumpel). The analysis was performed on four 3D image stacks. Shown images are aligned maximum intensity projections. Solid boxes indicate correctly identified spines, dotted boxes falsely identified spines and dashed boxes missing spines. The color of the boxes indicates the depth of the spine. Red lines mark axons (highlighted manually). Note that most spines are correctly detected despite significant background noise. (**B**) Spine detection in 3D in an image stack obtained in a different laboratory recording in a different brain structure using a different GFP label and microscope system (two-photon imaging of organotypic slice cultures from hippocampus; laboratory of A. Acker–Palmer).

of applying a neural network directly to raw images. While this method worked fairly well for the particular dataset used in this study, due to the predefined features the model appears more rigid. Moreover, the results were not compared with labels from multiple annotators. For these limitations, it is not clear whether this model generalizes well to other datasets. The approach by Rada et al.[24] has the advantage of traceability of the detection process. However, it is outperformed by the method proposed by Xiao et al.[35]. The latter study uses a method more similar to ours, but reduces the object detection task to detecting the center of spines, which neglects its shape. Nevertheless, our approach outperforms this model in terms of detection performance and speed, and we show generalizability to new, distinct datasets. While these previous methods apply neural networks to the task of spine detection, none of them does so by exploiting information across three dimensions. Xiao et al.[35] use 3D images but then calculate the maximum intensity projection. With our tool it is possible to flexibly switch between fast detections using maximum intensity projection and detailed three-dimensional tracking of dendritic spines.

The work by Argunşah et al.[37] introduces a comprehensive dendritic spine analysis GUI, which contains a CNN-based spine detection pipeline. The detection model is trained from scratch using large amounts of labeled training data, achieving a satisfying performance on their datasets. Testing their model on our data (without retraining), we obtained an F1-score of 0.44 for the majority ground truth. The model detects well even very weakly labeled spines. Being based on SURF features, the method relies mostly on local features of spine structure and, in our experience, appears to work best on circularly-shaped spines. Our model, owing to the fact that it is based on a pre-trained general purpose detection network, not only requires fewer labeled data for model training, but can be flexibly adjusted to new datasets, as it utilizes a broad range of local and non-local features for spine detection.

Recently, Fernholz et al.[44] proposed DeepD3, a spine segmentation method based on a U-Net-inspired architecture. For each 2D image, their model outputs both a binary segmentation mask for dendrites and a non-binary segmentation mask for dendritic spines. The latter is then post-processed to obtain either a 2D or 3D segmentation mask with labels available for each individual spine. This method achieves high values for recall. We evaluated DeepD3 on our data by counting segmented spines as true positives whenever their center of mass was inside a bounding box from our ground truth annotations (excluding multiple assignments to the same box). We found that the F_1-score on the majority ground truth was 0.22, achieving a high value of recall, but a relatively small precision due to false positives in axonal structures and background noise in our data. In our hands, this performance was a bit higher than that of DeepSpineTool[45], another recently proposed spine segmentation tool based on 3D U-Nets. However, directly comparing the performance of these models to one another and to that of ours, based on applying them to our data, could be problematic as they have not been trained on our data specifically, which tend to show a lower density and larger sizes of spines compared to the data in Fernholz et al. Unlike our method, these methods have the advantage of performing segmentation, which can be used, for instance, to compute geometrical features of dendritic spines.

As Fernholz et al. also provide a benchmark dataset, we were also able to assess the performance of our model (without further training or adjustment of hyperparameters) on their data in 2D. To this end we used the algorithm provided by Fernholz et al. to match the positions of spines predicted by our model (computed by the centers of mass of the predicted 2D bounding boxes) to those from their experts' annotations. The F_1-scores for our model, averaged over four runs, were 0.43 ± 0.07 and 0.47 ± 0.05 for the minimal and maximal ground truth, respectively. The drop in performance of or model relative to the performance using our own data may again be explained by the significant differences between the two datasets regarding spine size and density. However, the fact that the F_1-scores our model achieves on their data are still moderately high despite these strong differences in spine organization suggests some degree of robustness in our model, presumably as a consequence of the fact that it is based on a general purpose detection network.

In summary, we devised a new method for spine detection that is fast, accurate and relatively robust, and thus well suited for large datasets with thousands of spines. The method can achieve high detection performances, even when being applied to novel datasets obtained under different experimental conditions. Moreover, the method is flexible and can be further improved by anyone using additional suitable datasets for retraining the model.

Data availability

The data mentioned in this article, specifically the training, validation, and test dataset and the models used for evaluation, is openly available on GitHub at https://github.com/SaILaIDiN/Spine-Detection-with-CNNs.

Received: 14 February 2023; Accepted: 8 November 2023
Published online: 22 November 2023

References

1. Beaulieu, C. & Colonnier, M. A laminar analysis of the number of round-asymmetrical and flat-symmetrical synapses on spines, dendritic trunks, and cell bodies in area 17 of the cat. *J. Comp. Neurol.* **231**, 180–189 (1985).
2. Knott, G. W., Holtmaat, A., Wilbrecht, L., Welker, E. & Svoboda, K. Spine growth precedes synapse formation in the adult neocortex in vivo. *Nat. Neurosci.* **9**, 1117–1124 (2006).
3. Arellano, J. I., Espinosa, A., Fairen, A., Yuste, R. & DeFelipe, J. Non-synaptic dendritic spines in neocortex. *Neuroscience* **145**, 464–469 (2007).
4. Loewenstein, Y., Yanover, U. & Rumpel, S. Predicting the dynamics of network connectivity in the neocortex. *J. Neurosci.* **35**, 12535–12544 (2015).
5. Perkel, D. H. & Perkel, D. J. Dendritic spines: Role of active membrane in modulating synaptic efficacy. *Brain Res.* **325**, 331–335 (1985).
6. Murakoshi, H. & Yasuda, R. Postsynaptic signaling during plasticity of dendritic spines. *Trends Neurosci.* **35**, 135–143 (2012).
7. Holler-Rickauer, S., Köstinger, G., Martin, K. A., Schuhknecht, G. F. & Stratford, K. J. Structure and function of a neocortical synapse. *BioRxiv* 875971 (2019).
8. Holler, S., Köstinger, G., Martin, K. A., Schuhknecht, G. F. & Stratford, K. J. Structure and function of a neocortical synapse. *Nature* **591**, 111–116 (2021).
9. Lüscher, C., Nicoll, R. A., Malenka, R. C. & Muller, D. Synaptic plasticity and dynamic modulation of the postsynaptic membrane. *Nat. Neurosci.* **3**, 545–550 (2000).
10. Yuste, R. & Bonhoeffer, T. Morphological changes in dendritic spines associated with long-term synaptic plasticity. *Annu. Rev. Neurosci.* **24**, 1071–1089 (2001).
11. Day, M. *et al.* Selective elimination of glutamatergic synapses on striatopallidal neurons in Parkinson disease models. *Nat. Neurosci.* **9**, 251–259 (2006).
12. Graveland, G., Williams, R. & DiFiglia, M. Evidence for degenerative and regenerative changes in neostriatal spiny neurons in Huntington's disease. *Science* **227**, 770–773 (1985).
13. Grutzendler, J., Helmin, K., Tsai, J. & Gan, W.-B. Various dendritic abnormalities are associated with fibrillar amyloid deposits in Alzheimer's disease. *Ann. N. Y. Acad. Sci.* **1097**, 30–39 (2007).
14. Shankar, G. M. *et al.* Natural oligomers of the Alzheimer amyloid-β protein induce reversible synapse loss by modulating an NMDA-type glutamate receptor-dependent signaling pathway. *J. Neurosci.* **27**, 2866–2875 (2007).
15. Holtmaat, A. & Svoboda, K. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nat. Rev. Neurosci.* **10**, 647–658 (2009).
16. Birkner, A., Tischbirek, C. H. & Konnerth, A. Improved deep two-photon calcium imaging in vivo. *Cell Calcium* **64**, 29–35 (2017).
17. Meng, G. *et al.* High-throughput synapse-resolving two-photon fluorescence microendoscopy for deep-brain volumetric imaging in vivo. *Elife* **8**, e40805 (2019).
18. Loewenstein, Y., Kuras, A. & Rumpel, S. Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo. *J. Neurosci.* **31**, 9481–9488 (2011).
19. Rumpel, S. & Triesch, J. The dynamic connectome. *e-Neuroforum* **22**, 48–53 (2016).
20. Chambers, A. R. & Rumpel, S. A stable brain from unstable components: Emerging concepts and implications for neural computation. *Neuroscience* **357**, 172–184 (2017).
21. Caroni, P., Donato, F. & Muller, D. Structural plasticity upon learning: Regulation and functions. *Nat. Rev. Neurosci.* **13**, 478–490 (2012).

22. Yuan, X., Trachtenberg, J. T., Potter, S. M. & Roysam, B. MDL constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics* **7**, 213 (2009).
23. Janoos, F. *et al.* Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging. *Med. Image Anal.* **13**, 167–179 (2009).
24. Rada, L. *et al.* Tracking-assisted detection of dendritic spines in time-lapse microscopic images. *Neuroscience* **394**, 189–205 (2018).
25. Zhang, Y. *et al.* A neurocomputational method for fully automated 3D dendritic spine detection and segmentation of medium-sized spiny neurons. *Neuroimage* **50**, 1472–1484 (2010).
26. Bai, W., Zhou, X., Ji, L., Cheng, J. & Wong, S. T. Automatic dendritic spine analysis in two-photon laser scanning microscopy images. *Cytom. Part A: J. Int. Soc. Anal. Cytol.* **71**, 818–826 (2007).
27. Su, R., Sun, C., Zhang, C. & Pham, T. D. A novel method for dendritic spines detection based on directional morphological filter and shortest path. *Comput. Med. Imaging Graph.* **38**, 793–802 (2014).
28. Mostavi, M., Chiu, Y.-C., Huang, Y. & Chen, Y. Convolutional neural network models for cancer type prediction based on gene expression. *BMC Med. Genomics* **13**, 1–13 (2020).
29. Sekaran, K., Chandana, P., Krishna, N. M. & Kadry, S. Deep learning convolutional neural network (CNN) with gaussian mixture model for predicting pancreatic cancer. *Multim. Tools Appl.* **79**, 10233–10247 (2020).
30. Chen, T. & Chefd'Hotel, C. Deep learning based automatic immune cell detection for immunohistochemistry images. in *International Workshop on Machine Learning in Medical Imaging*, 17–24 (Springer, 2014).
31. He, T., Mao, H., Guo, J. & Yi, Z. Cell tracking using deep neural networks with multi-task learning. *Image Vis. Comput.* **60**, 142–153 (2017).
32. Kyme, A. *et al.* Markerless motion tracking of awake animals in positron emission tomography. *IEEE Trans. Med. Imaging* **33**, 2180–2190 (2014).
33. Nath, T. *et al.* Using deeplabcut for 3D markerless pose estimation across species and behaviors. *Nat. Protoc.* **14**, 2152–2176 (2019).
34. Wu, A. *et al.* Deep graph pose: A semi-supervised deep graphical model for improved animal pose tracking. *Adv. Neural Inf. Process. Syst.* **33**, 6040–6052 (2020).
35. Xiao, X. *et al.* Automated dendritic spine detection using convolutional neural networks on maximum intensity projected microscopic volumes. *J. Neurosci. Methods* **309**, 25–34 (2018).
36. Smirnov, M. S., Garrett, T. R. & Yasuda, R. An open-source tool for analysis and automatic identification of dendritic spines using machine learning. *PLoS ONE* **13**, e0199589 (2018).
37. Argunşah, A. Ö. *et al.* An interactive time series image analysis software for dendritic spines. *Sci. Rep.* **12**, 1–15 (2022).
38. https://github.com/SaILaIDiN/Spine-Detection-with-CNNs.
39. Dutta, A., Gupta, A. & Zissermann, A. VGG image annotator (via). http://www.robots.ox.ac.uk/vgg/software/via (2016).
40. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28**, 91–99 (2015).
41. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1492–1500 (2017).
42. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. in *European Conference on Computer Vision*, 740–755 (Springer, 2014).
43. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8026 (2019).
44. Fernholz, M. H., Guggiana Nilo, D. A., Bonhoeffer, T. & Kist, A. M. Deepd3, an open framework for automated quantification of dendritic spines. *bioRxiv* 2023–02 (2023).
45. Vidaurre-Gallart, I. *et al.* A deep learning-based workflow for dendritic spine segmentation. *Front. Neuroanat.* **16**, 817903 (2022).

Acknowledgements

We thank Dominik Aschauer, Noelia Mateu Fernández and Takahiro Noda for the annotation of spine data. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 414985841 GRK 2566 "iMOL" (M.K.), SPP 2041 "Computational Connectomics" (J.T., P.O.-V., S.R., M.K.), CRC 1080 "Neural Homeostasis" (A.A.P., D.B., S.R.), DFG/ANR project no. 431393205 (S.R.), and DFG DIP "Neurobiology of Forgetting" (S.R., M.K.).

Author contributions

S.R., D.B. and A.A.-P. conducted the experiments and provided the imaging data. S.R. and B.E. annotated imaging data. F.V. and M.K. developed the model. F.V. and S.A. wrote the computer code and evaluated the model with input from J.-B.E. and P.O.-V. F.V., J.T. and M.K. wrote the manuscript with input from all authors.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.