High precision automated detection of labeled nuclei in terabyte-scale whole-brain volumetric image data of Mouse

Girraj Pahariya*, Sukhendu Das*, Jaikishan Jayakumar*, Samik Banerjee*,
Venu Vangala*, Keerthi Ram†, Partha Mitra‡
*Computer Science & Engg. Dept., IIT Madras, Chennai, IN 600036
†Healthcare Technology Innovation Centre, IIT Madras, Chennai, IN 600036
‡Cold Spring Harbor Laboratory, NY 11724

***Abstract*—There is a need in modern neuroscience for accurate and automated image processing techniques for analyzing the large volume of neuroanatomical data. For e.g., the use of light microscopy to image whole mouse brains in a mesoscopic scale produces individual neuroanatomical data volumes in the TerraByte range.. A fundamental task involves the detection and quantification of objects of a given type, e.g. neuronal nuclei or somata, in whole mouse brains. Traditionally this quantification is performed by human visual inspection with high accuracy, that is not scalable.. When state-of-the-art CNN and SVM-based methods are used to solve this classification problem, they achieve accuracy levels between $85 - 92\%$. However, higher rates of precision and recall, close to that of humans are necessary. In this paper, we describe an unsupervised, iterative algorithm, which provides a high close to human performance for a specific problem of broad interest, i.e. detection of Green Fluorescent Protein labeled nuclei in whole mouse brains. The algorithm judiciously combines classical computer vision (CV) techniques and is focused on the complex problem of decomposing strong overlapped objects (nuclei). Our proposed iterative method uses features detected on ridge lines over distance transformation and an arc based iterative spatial-filling method to solve the problem. We demonstrate our results on two whole mouse brain data sets of Gigabyte resolution and compare it with manual annotation of the brains. Our results show that an aptly designed CV algorithm with classical feature extractors when tailored to this problem of interest achieves near-ideal human-like performance. Quantitative analysis, when compared with the manually annotated ground truth, reveals that our approach performs better on whole mouse brain scans than general purpose machine learning (including deep CNN) methods.**

***Index Terms*—GFP, cell-detection, RCNN, convex arcs, mouse brain, mDT, ridge.**

I. Introduction

Understanding the cytoarchitecture of the whole brain is an important goal in the field of neuroscience [21], [44]. In particular, the identification of locations and quantification of neurons across the whole brain is a vital task. To achieve this goal, modern imaging methods have allowed users to produce images at very high resolution for analytics. Entire mouse brains can now be imaged using light microscopy at sub-micron resolution from entire mouse brains ($\sim 1cm^3$ volume), with individual data volumes in the Gigabyte range. Optical microscopic techniques applied at such high resolutions, help to study the whole brain at a single neuron resolution [31], [36] but the large terabyte individual data volumes make it a significantly computational and data analytics challenge.

In addition to the advancements in imaging, advanced methods such as the expression of the green fluorescent protein (GFP) gene [9] have ensured that specific types of neurons can be tagged and studied in great detail [46]. A successful implementation of GFP tagging in the identification and quantification of interneurons is described in [46]. Manually studying the distribution of these GFP expressing neurons across the entire brain is often time consuming and is susceptible to human errors over large datasets. To overcome this, an automated process of detection and quantification of neurons [3], [23] must be designed based on Image Processing (or Computer Vision) algorithms. However, the level of sophistication in the analysis techniques have not yet caught up with the significant improvements in microscopic imaging methods. In this work, we have sought to develop an automated method of detection for GFP tagged interneurons across a whole mouse brain. Thus, the aim of our work, proposed in this paper, is to detect and count GFP tagged specific cell types from entire mouse brains.

Automated algorithms are often resource consuming and have been deployed in studying the brains of lower species such as Candida elegans [47]. Computer vision algorithms have also played a vital role in the segmentation of large-scale electron microscopy data. Kaynig *et al.* [22] have proposed a pipeline that provides state-of-the art reconstruction performance while scaling to data sets in the GB-TB range. Such high performance was acheived by training a random forest classifier on interactive sparse user annotations. This classifier output was then combined with an anisotropic smoothing prior in a Conditional Random Field framework to generate multiple segmentation hypotheses per image. These segmentations were then combined into geometrically consistent 3D objects by segmentation fusion. A more recent work in automatic neural reconstruction from petavoxel of Electron Microscopy data [45] suggested a dense Automatic Neural Annotation framework (RhoANA) to automatically align, segment and reconstruct a $1mm^3$ brain tissue (~ 2 peta-pixels), using a web-based tool to manually proofread the output, and ensure reconstruction correctness. The pipeline performs membrane classification and 2D segmentation. using state-of-the-art deep learning (DL)

techniques in order to generate membrane probability maps.

Deep learning methods have been used in the detection of cells in cancer biology. One method for mitosis detection [2] is based on DL which performs data aggregation using CNN via additional crowdsourcing layer (AggNet). Mualla *et al.* [32] propose a method for automatic cell detection on polypropylene substrate, suitable for microbeam irradiation. A Harris corner detector was employed to detect apparent cellular features. These features-corners were grouped based on a dual-clustering technique according to the density of their distribution across the image. Weighted centroids were then extracted from the clusters of corners and constituted the targets for irradiation. Quelhas *et al.* [39] introduce a new approach for cell detection and shape estimation to detect overall convex shapes in multivariate images based on a sliding band filter (SBF). Furthermore, the parameters involved are intuitive as they are directly related to the expected cell size. Using the SBF filter, cell nuclei, the cytoplasm location and shapes were detected. In the remainder of the paper, we will refer to the terms cells and GFP nuclei interchangeably (as our target objects).

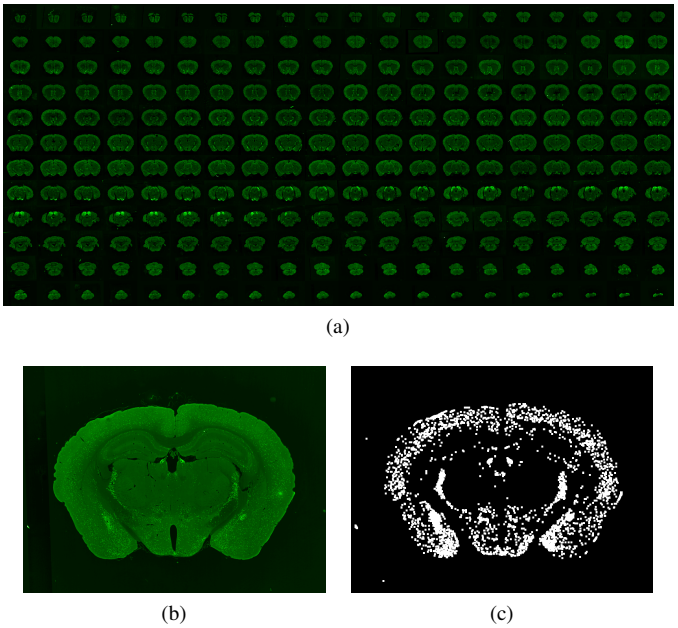


Fig. 1: shows (a) 250 individual sections obtained from a whole mouse brain (Hua-166); (b) Single example section from the dataset shown in (a) demonstrating the GFP tagged areas within the brain; (c) example of the extracted Green channel data from (b).

Figure 1 shows an example of the dataset for a whole mouse brain (*Hua-166*). The dataset consists of 250+ sections that have been scanned at $18K \times 24K$ resolution at $0.46 micron per pixel$ scale. Figure 2 shows an example of a cropped region from one section (figure 1(b)) where the object-detection methods based on RCNN [19] and SVM [13], [33] have been deployed. As demonstrated in 2(c) and 2(d), these methods, although fairly accurate still produces some false alarms (detections) and misses, especially when detecting strong overlapping objects. As a scientific and diagnostic tool, this level of accuracy is not acceptable to pathologists and neuroscientists and higher rates of precision and recall are necessary. The major reason for the moderate accuracy given by latest supervised methods is that, various non-unique patterns of spatial layouts and structures are formed by clusters of neurons, specifically with strong overlap. Machine learning algorithms [5], [16] have been designed to produce high accuracy invariant to pose, scale (size) background and texture of the object. However, large amount of overlap (dense crowd, large clutter/pile of objects, say) causes failures due to occlusion and camouflage. Deep-CNNs have yielded rich dividends in example cases, ranging from IMAGNET [15], PASCAL [17] image datasets, to HMDB [24], Hollywood-II [28], MS-COCO [25] video datasets, where considerable inter-class variability exists in the data, unlike in our dataset. Moreover, none of these public datasets contain objects with large amount of clutter and strong overlap, which is the case in some scans of our brain dataset.

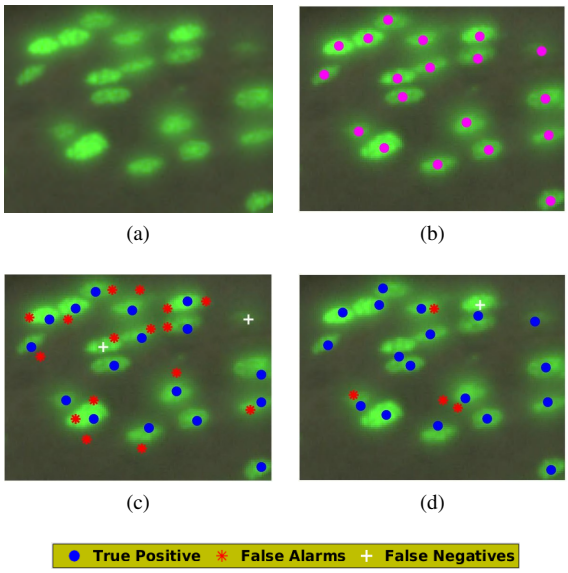


Fig. 2: Examples of application of SVM and RCNN for the detection of GFP tagged nuclei on a cropped region of mouse brain section (a) cropped sample from a whole brain section; (b) Ground truth of cell centers marked in magenta dots; Results of cell detection using: (c) SVM [13] (Gaussian kernel) & (d) RCNN [19] on (a).

Detection of cell centers in cases of strong overlap of objects is a difficult problem, as the overlapping objects result in complex 2D structures that do not exhibit a unique appearance pattern of shape, size and overlap formation (layout) that can be trained accurately by any shallow or deep learning (DL) methods. However, it is necessary for any automated algorithm to achieve high accuracy close to that of human annotators, but over a large scale ($\simeq$ 400 GB size images) and lesser computational time (few minutes on a single high-end CPU machine). With this aim, we have proposed an improved and efficient algorithm for the unsupervised classification of GFP tagged GABAergic neurons across the whole mouse brain [6]. Our proposed method is designed suitably as a congruence of image analysis algorithms, such as Distance Transforms (DT) and edge detection to quantify the boundaries of the detected cells, followed by the extraction of features along ridge lines for detecting the candidate centers of neurons. The algorithm

is further enhanced by an iterative approach of region filling by circular blobs of adaptive size, using convex arcs on the edge maps for detecting overlapped cell areas. Using this method, we have obtained a high precision of 0.972 and a recall of 0.995, over the two complete whole brain section image sets (~ 250 scans in each set). A comparative performance analysis shows that our proposed algorithm surpasses two existing deep learning Convolutional Neural Networks (CNN) models [19], as well as two popular shallow learning object detection algorithms. Shallow and deep models are not tailor-made for a structure decomposition problem which is an essential component necessary for detecting overlapping objects (GFP tagged nuclei in our case).

II. Overview of the Proposed Method

Individual sections of the mouse brain are obtained as gigapixel resolution (RGB $18K \times 24K$ pixels at 12-bit channel) images, from the whole mouse brain of the wild type mouse (C57BL6), with green fluorescent protein (GFP) tagging of GABAergic neurons (PV+) [1]. This data is part of the mouse brain architecture project [1] at the Cold Spring Harbor Laboratory (CSHL), which aims to quantify the connections of different parts of the brain using tracer injections and cell tagging techniques at a mesoscopic scale [6]. The CRE cell lines that tag interneurons were developed by Josh Huang's Laboratory and have been described in [46]. The histological and imaging preparation of the brain has been previously described in [38]. Manual annotation of two sets of brains were performed by researchers from the Indian Institute of Technology (IIT), Madras, India, making the comparison and evaluation of algorithms feasible on the scan data. Details of the manual annotation is described in the later results section.

Figure 3 shows two example sections of GFP tagged neurons in a mouse brain with different distributions (as density of occurrence) - minimal overlap and strong overlap. For clarity, the images in figure 3 are cropped parts of large gigapixel resolution scans of mouse brain. The GFP tagged cells appear with significant intensity in the green spectrum of the visible light within the microscopic scans. Objects (cells) in the overlapping regions appear as a dense 2D packed layer, where more than 3 cells overlap (figure 3(b)). Individual nuclei in these regions also appear with varying sizes within the regions of interest. This arbitrary spatial arrangement of cells with significant overlap make it a challenging task to detect each one of them accurately. The unsupervised computer vision approach of cell detection proposed in this paper has been designed and targeted for dealing precisely with this situation of detecting packed overlapping nuclei, using geometric constraints of the cell layout to decompose a structure into overlapping circular units.

A. Basis and Justification of the algorithm designed

For our problem scenario, the solution is to identify centers of cell boundaries. We have achieved this solution by using Edge detection, Distance Transform and Hough transform as default processes of *Stage I* in our pipeline. This part of the designed process accurately detects cells in simple cases, where the GFP tagged nuclei are distributed in a sparse or

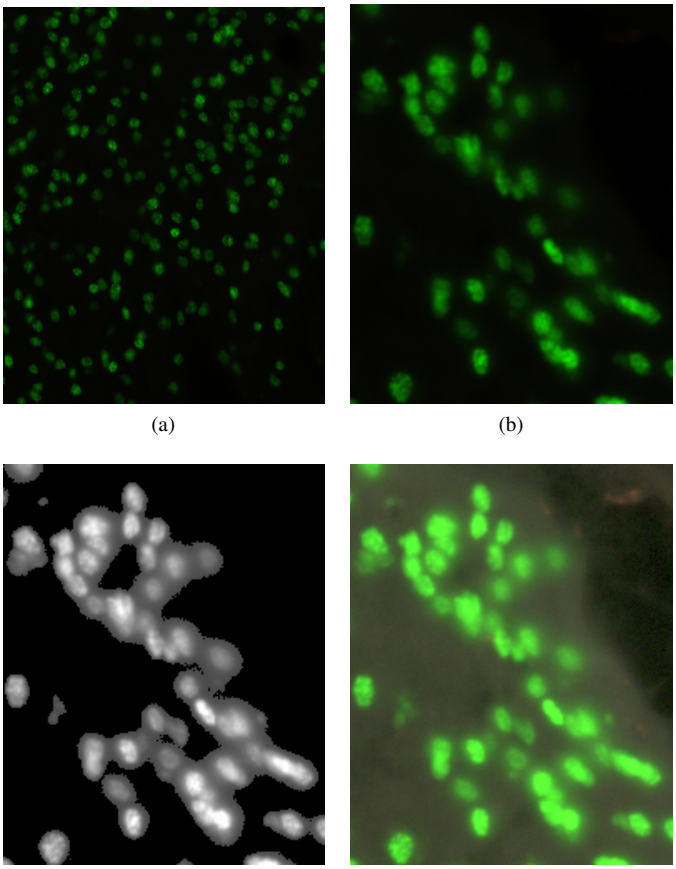


Fig. 3: Examples of distributions of GFP tagged interneurons in a mouse brain, for (a) Minimal overlap, (b) Strong overlap of cells; (c) Salient foreground region (FGL) obtained by thresholding the green channel of (b); and (d) the direct histogram specification (DHS) of (c) (best viewed in color).

minimal overlap manner. In case of overlap of multiple cells(as seen in figures 2 and 3(b)), a separate process (*Stage II*) is required for accurate identification of the nuclei.The solution for such complex cases is hard and fails if conventional machine learning models are used. The process we have designed uses geometric constraints with area filling to decompose a 2-D area of overlapping cells into partly occluded cell bodies. Thus, geometric constraints to fill cells in overlapping cases is used to solve the complex cell detection task. This fundamental step makes our algorithm's performance to be better than deep-CNN and shallow ML methods. The complexity of the data lies in the degree of overlap of objects within the image and is dependent on the type of object,the microscopic and sample preparation techniques. The DL models, although well suited for multiple object detection are not well suited to the overlap-decomposition problem which we have implicitly addressed in our pipelined process. The simplicity of method when combined with close to human performance, makes it advantageous to use over other methods including CNNs for solving this problem.

B. Brief Description of the proposed algorithm

As the GFP tagged neurons have relatively higher irradiance in green channel compared to the background neuropil, we have enhanced the 12-bit green channel (see figure 3(b) as an

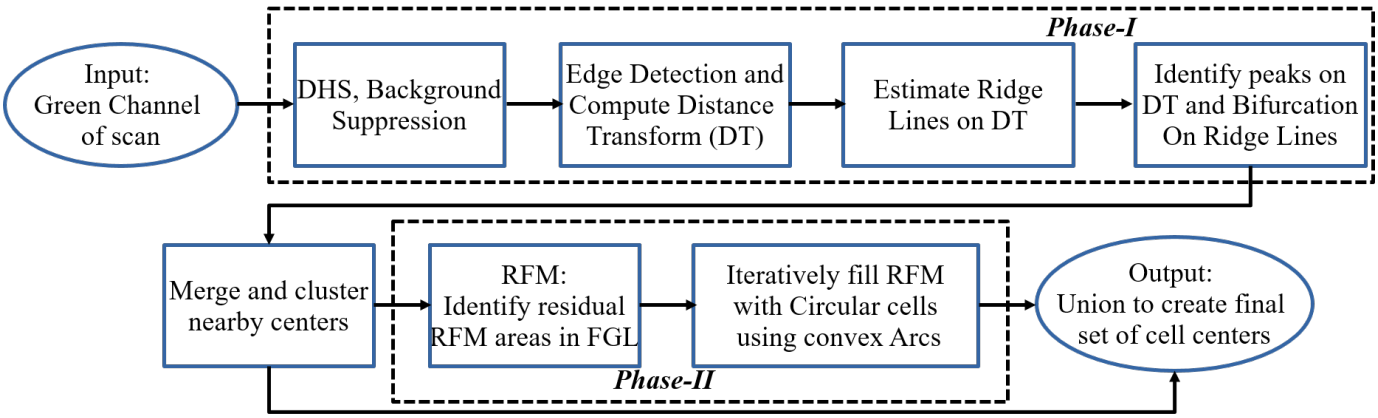


Fig. 4: Flowchart detailing the algorithmic flow of the proposed method for the detection of centers of GFP tagged nuclei in whole mouse brains. *Phase I*: Method based on Distance Transform and ridge based features; *Phase II*: the iterative arc based method of area-filling. A more detailed outline of *Phase II* is given in figure 6.

example) using Direct Histogram Specification (DHS) [42] as a pre-processing step. Output of applying DHS on the sample in figure 3(b) is shown in figure 3(c). Foreground segmentation by thresholding the DHS output (figure 3(c)) produces the salient foreground region (i.e., FGL) as shown in figure 3(d). It was observed that modern machine learning (ML) and deep learning (DL) based supervised object detection algorithms fail to reach the desired level of high accuracy in case of overlapping (figure 3(b)) arrangement of cells (illustrated in supplementary sections S.4 and S.5). This task is analogous to the problem of packing circles of varying (but within a limited range) sizes within an arbitrary polygonal shape. Simpler versions of this problem have been addressed in the fields of Computational Geometry and Operational Research [26], [41], [20], [29], [12], [8], [49], [27], [14]. This is an NP-hard problem [11] and no prior work exists to give a direct solution for such a task, specifically in cases of overlap of circles. Hence, one has to resort to either an iterative method based on optimization with certain constraints for an approximate solution, or use image features first to place few cells suitably at locations with high feasibility and then fill gaps in an iterative manner. We chose the latter path for our method , as the FGL map (in figure 3(d)) provides rich clues to initially detect a few cells with high level of accuracy.

Our proposed method consists of two sub-processes (illustrated in figure 4): *Phase-I* sub-process consists of a pre-processing step, which enhances and segments the foreground areas (FGL). This is followed by feature extraction, using the Distance Transform (DT) from edge map of FGL and ridge lines from the modulated DT (mDT) map. Peaks of mDT map and features on ridge lines correspond to candidate centers of detected objects of interest. *Phase-I*I sub-process fills the cell areas (detected centers with edges) detected in *Phase-I* with the background color on FGL to obtain a residual map. Significant areas of Residual Foreground Map (RFM) are then iteratively filled (using Hough Transform) with cells, using convex arc segments that were identified on the outer boundary edges of the residual foreground map. The output of the cell center locations provided by the two phases are merged using a union operation (spatial locations of nearby cell centers are averaged) to create the final set of detected cell centers. Both these phases of the pipeline (unsupervised) process are described below.

III. Phase-I: DT and Ridge based method for cell center detection

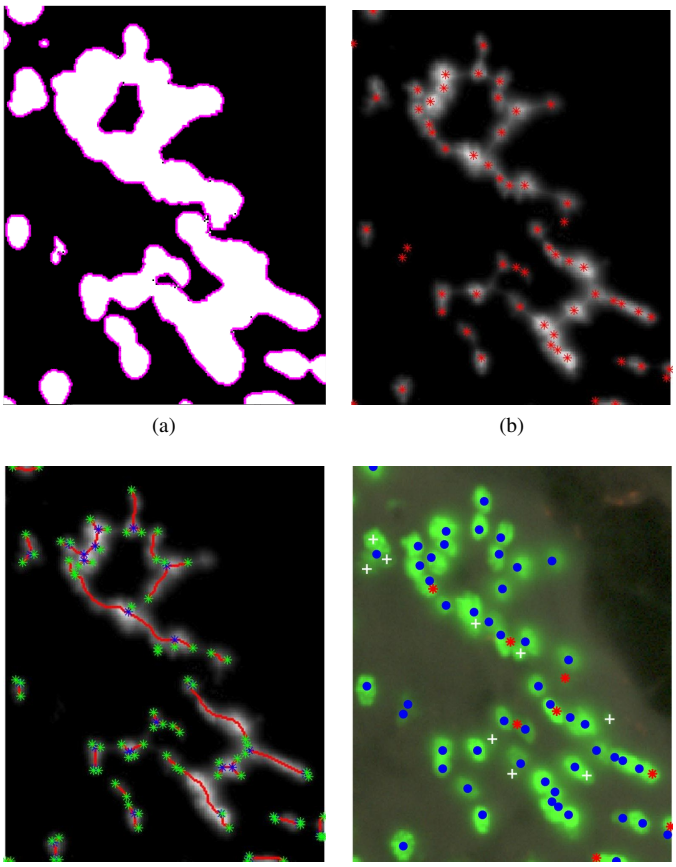
Salient regions of the green channel correspond to the nuclei of cell bodies. Since this is our main objects of interest, we have used the green channel alone for further processing. The process, however is applicable for any color channel. Steps used in Phase-I of our algorithm(see figure 4) are:

- Direct Histogram Specification (DHS) enhancement
- Threshold to get FGL
- Edge detection on FGL map
- Distance Transform (DT) on edge map
- Compute mDT map using DT and FGL
- Peaks and ridge line detection on DT map
- Detection of Bifurcation Points (BF) and Ridge Endings (RE) on the ridge lines
- Clustering of the peaks, BF and RE to yield the list of putative centers of cell nuclei

As a first step, we have enhanced the green channel using DHS (figure 3(c)), and used a thresholdi (TH, compared with green intensity) to produce the foreground layer (FGL) in the image (figure 3(d)). We then applied a Canny edge detection method [7] on the FGL to obtain an Edge map. A distance transformation (DT) [30] on the edge map extracted from FGL is then used to produce the putative boundaries of the detected cells. The pixel values at the background are then suppressed to zero, leaving only the DT-map on pixels of FGL. Figure 5(a) shows binary (thresholded) map of FGL in figure 3(d), with the edge map overlayed and represented by the Magenta borders. Supplementary figure S.1(b) shows the DT map extracted from the edge map of figure 5(a).

The DT map [37] results in producing an intensity peak which corresponds to the center of a detected nuclei, and this intensity tends to decrease as we get close to the edges, making the identification of isolated or partial overlap of two or more nuclei easier. However, when several GFP nuclei overlap, DT map does not always exhibit sharp peaks, making it difficult to detect cells using only a local maxima. Hence we have used a modulated DT (mDT) map obtained as:

$$g(x, y) = d(x, y).f(x, y), \forall(x, y) \tag{1}$$



(a) (b) (c) (d)

Fig. 5: (a) Binarized smoothed image obtained from the example in figure 3(d) overlayed with the edge map (in magenta) obtained using Canny edge detection algorithm; (b) peaks detected on the mDT map obtained from the product of (a) and the distance transform map; (c) Landmarks such as Bifurcation points (BF) and Ridge Endings (RE) on Ridge lines obtained using the modified "vessel" filter [18] algorithm, which corresponded to the centers of detected locations of potential cell nuclei; (d) Results by combining centers in figures 5(b) and 5(c), shown along with the true positives, false alarms and false negatives (markers identical to figure 2; best viewed in color). Results of the intermediate steps of this process are given in Supplementary section (S.1).

where, g, d and f represent mDT-map, DT value and FGL respectively, x, y represent integer pixel coordinates. The mDT map exhibits better localization of peaks as cell centers (figure 5(b)). Regional maxima in mDT map are hence detected and stored to represent the initial list of putative centers of the GFP nuclei (figure 5(b)).

This process described so far though works well in cases where the GFP nuclei are spatially distributed with minimal overlap, however produces unsatisfactory results when few cells have strong (large) overlap in arbitrary spatial arrangements. As closely packed clusters of cells do not follow any well-defined structure or geometrical arrangement, expecting the center-line (as in MAT [10]) to be the loci of peaks would be an erroneous assumption. In such cases, the maxima in mDT map may be falsely detected to signify a single center when there are actually two or more centers present in the FGL (figure 3(c)) for that location. This process of detecting peaks in mDT, thus results in a few "misses" of cell centers for the complex cases of strong overlap of cells.

To overcome this, we have designed an algorithm that uses the detection of features within the ridge lines extracted from mDT map and assigns the centers of the nuclei based on the location of these features on Ridge lines. This ridge detection process is similar to the "Vessels" Algorithm [18], that has been used earlier to detect blood vessels from magnetic resonance imaging (MRI) data. The features used in our analysis are the Bifurcation (BF) points and Ridge Endings (RE) of ridges. The Bifurcation points and Ridge ending were obtained as landmark features by traversing along the ridge lines (as in [42]). The spatial locations of the BF and RE correspond to the detected centers of potential GFP tagged nuclei. An example of those feature points detected on ridge lines are shown in figure 5(c). The pair of landmarks along a ridge could be any combination of either BF or RE, with the number of pixels lying between a pair of adjacent landmarks (RE and/or BP) on the ridge lines considered to be the gap (geodesic distance: GC). If the gap GC is large (greater than an empirical threshold, $L = 20$ pixels, the average size of well isolated detected nuclei), a set of cells are inserted at uniformly spaced locations between the corresponding pair of landmarks on the ridge. The number of cell centers inserted on a ridge line between a pair of adjacent features, is GC/Cell-diameter (CD), if there are no peaks detected apriori (see figure 5(b)) by the *Phase-I* of processing in the neighborhood of these newly identified locations on the ridge. This is the first geometric constraint used to detect a sequence of adjacent cells (partly overlapping) along a ridge line. Traversing along the ridge lines, as shown in figure 5(c), produces an additional list of centers inserted on ridge lines with large gap (GC) between a pair of landmarks. These centers estimated on the ridge lines and those obtained from the peaks (figure 5(b)) on mDT, are combined together (mostly union of two sets, and merging of nearby peaks) to yield the final set of cell centers at end of *Phase-I* of processing. Figure 5 (d) shows the results of intermediate level processing at the end of *Phase-I* (DT + Ridge-based method).

IV. Phase-II: Convex-Arc Based Iterative Method for Missing Cell Detection

Results of *Phase-I*, though fairly accurate, are not at par when compared with human performance particularly in case of large overlap of cells. An example of this failure is highlighted by figure 5(d) where true detections, misses and false alarms of *Phase-I* are shown alongside true centers as annotated by a master annotator.

To further enhance the efficiency of the algorithm, we have introduced an iterative process (figure 6) of cell filling using boundary arcs on a residual foreground layer (FGL in figure 3(c)). This process is based on a hypothesis of topological/geometric constraint: in case of strong overlap, cells at the periphery of the region of overlap produce the boundary (edge) layer of FGL. Those in the inner cordon fill up the gap to produce prominent green fluorescence within the

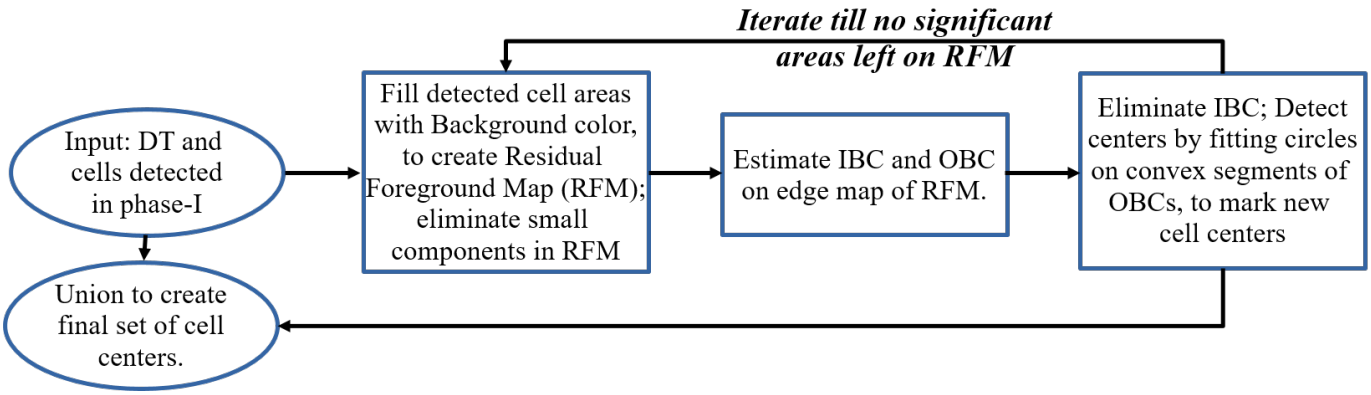


Fig. 6: Iterative arc based method for cell detection. This phase uses results from *Phase-I* of the algorithm as inputs and the final result of nuclei centers is the combination of the output from this phase and the *Phase-I*.

FGL. To efficiently detect the overlapping cells, an iterative process is designed to detect overlapping cells in FGL, starting with boundary layer, which is described below as algorithm 1.

Algorithm 1: Detecting Overlapping cells at *Phase-II*.

Input: The background suppressed attenuated FGL map (figure 3(d)); mDT map (figure S.1(b)); and the detected nuclei from *Phase-I*.

Output: Detected Overlapping cells.

```
while Significant foreground area left in RFM do
    Segregate the connected components which are greater than the average size of cells
    Suppress the regions of detected cells with background color to produce/update residual map (termed as Residual Foreground Map, RFM)
    Filter RFM by eliminating small connected components with an area less than a threshold A (30 pixels), leaving a few significant components in the RFM
    Segregate edge maps of these significant areas of RFM, based on convexity/concavity of edges, as: Inner Boundary (IBC) and Outer Boundary Contours (OBC)
    Ignore IBCs and select the OBCs
    Fit circles using Hough transform [40] on every convex arc between an adjacent pair of zero-crossings (ZCs) on the edge contours of RFM
end
```

This process (unsupervised) iteratively identifies undetected cells over strong overlapped regions; specifically over significant RFM areas, which are missed at the *Phase-I* of processing.

Various stages of the iterative process within the *Phase-II* of the algorithm are described in (figure 6). This method using cell-filling (fitting circles) by convex arcs on boundaries of FGL (Foreground Layer) is explained using a simulated example in figure 7. Figure 7(a) shows synthetic example of 4 overlapping cells, given as input to *Phase-I* of processing. Results of various intermediate stages are shown in the rest of the figure. Only the outline of the cell-boundaries are shown in this synthetic example. The steps of the process analyzing the example as given in figure 7, are given in supplementary section (S.2).

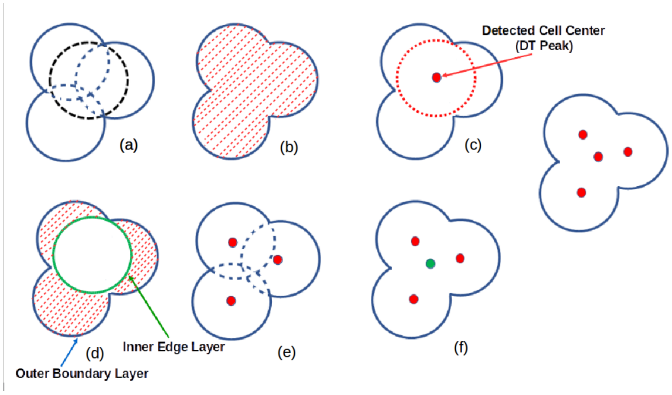
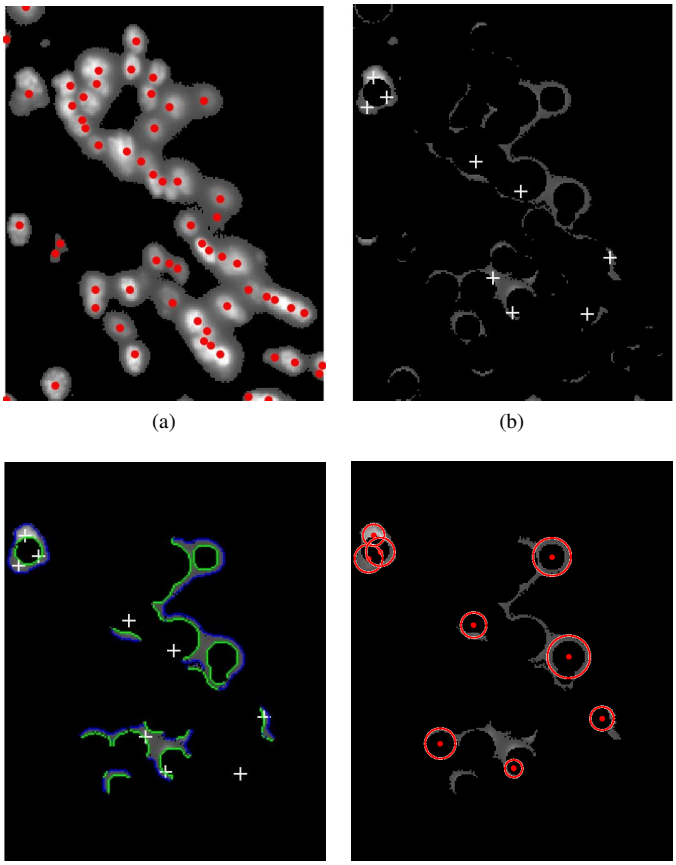


Fig. 7: A synthetic example to illustrate the processing steps of *Phase II* of the algorithm: (a) Simulated overlapping cell regions, (b) Edge map of the overlapping cell regions with FGL shaded; (c) detected cell center (using mDT Peak) and cell region, (d) Residual area, RFM, after suppressing the cell regions (detected in (c)); also, Inner Boundary Contour (IBC) marked in green and Outer Boundary Contour (OBC) marked in blue, (e) Circle fitted on the convex arcs or OBC using Hough Transform (HT), followed by extension to the rest of the circles shown in dashed curves and centers; (f) new cells detected in red along with center detected in green from (c); and (g) final set of detected centers on the synthetic example of strong overlap of cells in (a).

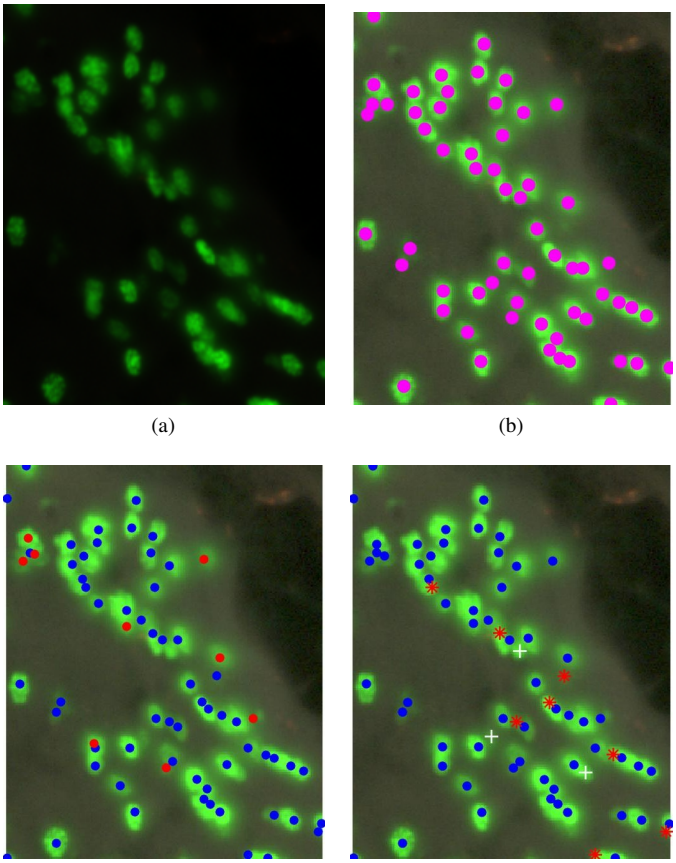
Figure 8 demonstrates the application of the iterative arc based method for missing cell detection in the given example from figure 3(d). Figure 8(a) shows the peaks detected at *Phase-I*. Figure 8(b) gives residual map (RFM) obtained after suppressing the detected cell regions in figure 8(a) with the background color. Figure 8(c) shows Inner boundary contours (IBCs) in green and Outer boundary contours (OBCs) in blue on the significant RFM after filtering out the small connected components from that in figure 8(b). IBCs are eliminated and convex arcs are fitted (using Hough Transform) with circles on convex OBCs. Figure 8(d) shows the centers of the new cells detected at *Phase-II* of processing.

The final list of cell centers is obtained by a merging process of the two lists of cell centers detected from *Phase I*: DT +



(a) (b) (c) (d)

Fig. 8: Application of *Phase II* on an example section containing strong overlap of GFP nuclei. (a) Centers detected by *Phase-I* of processing on FGL, as given in figure 5(d); (b) RFM (Residual foreground map) obtained after first iteration, from (a); (c) IBC in green and OBC in blue with missed cells on the significant RFM obtained from (b); and (h) cell centers detected by fitting circles using Hough Transform on convex OBCs (best viewed in color).



(a) (b) (c) (d)

Fig. 9: Comparisons between (a) the cropped section of a brain scan exhibiting a strong overlap scenario; (b) The ground truth obtained by manual annotation on (a); (c) Cell centers (GFP nuclei) obtained after merging centers obtained from *Phases I* and *II* of processing on (a), shown in blue and red dots respectively; and (d) gives true positives, false alarms and false negatives. (best viewed in color, marker coding follows that given in figure 2).

Ridge-based peak feature detection (*e.g.* figure 5(d)) and *Phase II*: Arc based iterative filling (*e.g.* figure 8(d)) methods. If two or more distinct cell centers detected at *Phase-I* and *Phase-II* stages fall within a small neighborhood ($dc < 2 * RC$, RC= average radius of a cell), they are merged and the average of the locations of the cell centers replaces the set of nearby cells. Figure 9 gives an illustration of the results of this process of merging and detecting new cells in residual foreground layer.

V. Results and Performance Analysis

We have evaluated our algorithm on GB resolution brain images of the mouse brain. The data [1] was manually annotated by a set of annotators ($n = 12$; 10 naive and 2 master annotators) to obtain the groundtruth. Each gigapixel resolution image was split into tiles of 500×500 pixels for easier quantification by the human annotators. Each annotator was also given the classification results of the DT and ridge based method (*Phase-I*), and then allowed to modify, insert or delete the existing cell centers. The output of the *Phase-I* was provided to reduce the burden of identifying all isolated cell locations by visual search. This did not bias their judgment in the complex regions of overlapped cells. Annotators easily modified (delete, insert, change location) the detected cells using a GUI (see snapshots in S.6 of the Supplementary Document). Providing manual annotators the output of a first-pass algorithm is "industry standard" in related problems (*e.g.* in EM for image segmentation). The annotation labels were cross-verified by a master annotator, which gave us confidence of its accuracy levels.

Figure 10 shows typical examples of cell detection for selected parts of mouse brain scans: minimal and strong overlapping regions respectively, obtained using our proposed method. The proposed algorithm takes the cell centers detected from two phases, *Phase-I*: DT + Ridge-based; and *Phase-II*: Arc-based methods, and combines them using an iterative combination strategy. True positives, false alarms and false negatives are marked with blue, red and white color markers respectively in the outputs. *Phase-I* sub-process is sufficient to detect all cells accurately in sparse and minimal overlap regions. The *Phase II* sub-process requires $2-3$ iterations to converge,

typically in cases of strong overlapped cells. The details of training the supervised architectures are given in the next subsection, and are elaborately discussed in the Supplementary document (S.4 for RCNN and SegNET).

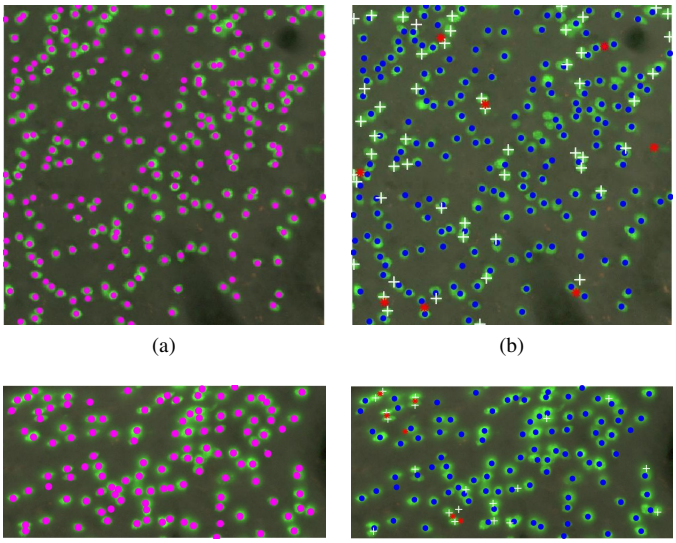


Fig. 10: Result of cell detection process using proposed method (figure 4); (a) and (c) show regions with minimal and strong overlap of cells respectively with ground truth; (b) and (d) are results evaluated from our proposed method corresponding to scans in (a) and (c) respectively (best viewed in color). See figure 2 for color marker coding.

A. Evaluation of the proposed algorithm

The manually annotated samples were used as groundtruth to evaluate the performance of the algorithm. In addition, the performance of our proposed method was compared with other methods that can potentially be applied to solve the same problem. These were: (a) a simple and fast unsupervised method of cell detection using Gaussian convolution; (b) *Phase-I* of our proposed approach (for comparing with *Phase-II*); two supervised shallow learning methods: (c) SVM [13], [34] (using Gaussian Kernel); and (d) HAAR + Adaboost [48]; and two DL techniques: (e) RCNN [19] and (f) SegNet [4]. For the supervised approaches, we have used half of the annotated data for training and the remainder of the data for testing. A Precision-Recall metric were used as the performance criteria [35]. The cell-centers that were detected within a certain distance ($DR < 5$ pixels) of true location in groundtruth were considered as to belonging to the same cell. Cell-centers of positively classified windows are considered as detected cell centers. The details of training the supervised architectures elaborately discussed in the Supplementary document (S.4 for RCNN and SegNET).

We have used a tensor flow based implementation of RCNN [43] for our experiments. RCNN [19] performs a bounding box regression on the objects in an image for detection. We have used equal number of samples for training and testing. Since we have only one object of interest *i.e.* GFP labeled neuron in our data, we provide a bounding box of size 25×25 pixels (derived empirically) around each cell. RCNN [19] uses the coordinates of the positive samples as the anchor boxes. Centroids of the bounding boxes obtained as output are considered as detected cell centers. We have utilized a SegNet [4] model that has been trained as per the code provided by the authors. Training samples (only positive ones) in SegNet were generated by delineating circular patches around manually annotated cell centers. The inputs for training were circular patches of 4 pixels radii around the manually annotated cell-centers.. The trained SegNet map produces outputs as circular patches (sometimes overlapping with each other) on the test image, whose centers can be considered as putative cell-centers. Details of training are given in Supplementary section (S.3).

Comparative analysis of our algorithm with the other models are tabulated in table I. Our iterative approach exhibits high precision and recall values of 0.972 and 0.995 (average over two complete datasets of whole mouse brains) when evaluated using the manually annotated ground truth, indicating the close to human efficiency of the algorithm, the Gaussian smoothing approach is faster compared with other methods and also provides a high recall value. The supervised methods of cell detection based on SVM and RCNN-based deep learning did not perform better than our proposed unsupervised approach.

Algorithms for cell detection	**Hua**	**166**	**Hua**	**167**
	Pr	**Rcl**	**Pr**	**Rcl**
Gaussian Convolution	0.883	0.979	0.825	0.946
Phase I: DT based method (figure 4)	0.966	0.983	0.964	0.987
SVM based detection [13], [34]	0.912	0.892	0.923	0.898
HAAR + Adaboost [48]	0.896	0.885	0.892	0.891
RCNN [19]	0.893	0.906	0.908	0.913
SegNet [4]	0.853	0.841	0.879	0.886
Overall proposed method (figure 4)	**0.974**	**0.997**	**0.968**	**0.992**

TABLE I: Performance of our algorithm averaged over two sets of whole mouse brains (251 scans of *Hua-166* and 244 scans of *Hua-167*, with $18K \times 24K$ pixels each). Our approach provides the best precision and recall over supervised machine learning (shallow learning with 5-fold validation and deep learning methods) approaches. ***Pr*** - Precision ; ***Rcl*** - Recall.

Our proposed unsupervised iterative method of cell detection, provide near-ideal performance in most of the cases, with exceptions that can be attributed to the presence of noise in the dataset. In order to demonstrate the effectiveness of the algorithm over wide distribution of cell nuclei throughout the brain, we have separated the results over strong overlap and minimal overlap and sparse regions within cells of the brain, and results for both sub-processes are given in table II. As expected, the algorithm performs the best in the sparse regions with marginal degradation in performance in the minimal overlap and strong overlapping areas. Table III shows the comparative performance, on the separated data sets of overlapped, sparse and minimal overlap regions of the brain, of our proposed method compared to that of SVM [13] and RCNN [19], [43]. Additional results (with examples) of cell detection of the competing supervised methods used for performance

analysis are given in Supplementary section (S.5).

Performance our proposed method on Cell distributions in Mouse brain scans	**Hua-166**		**Hua-167**	
	Pr	**Rcl**	**Pr**	**Rcl**
Strong Overlap of multiple cells, in parts of $\sim$ 5 scans/brain (Overall)	0.952	0.97	0.94	0.98
Strong Overlap of multiple cells, in parts of $\sim$ 5 scans/brain (Phase-I)	0.86	0.74	0.85	0.78
Minimal overlapped Cells, in parts of $\sim$ 20 scans/brain (Overall)	0.96	0.99	0.95	0.97
Minimal overlapped Cells, in parts of $\sim$ 20 scans/brain (Phase-I)	0.89	0.78	0.90	0.84
Sparse Cells, in abundance $\sim$ 224 scans/brain (overall/*Phase-II*)	0.978	1.0	0.97	0.99

TABLE II: Performance of proposed unsupervised method evaluated separately on the different sections of the brain, with strong overlap and minimal overlap of cells, for the proposed method; **Pr** - Precision ; **Rcl** - Recall.

The proposed iterative approach (*Phase II*) converges in typically $2-3$ iterations for the complex overlap cases. For the example shown in figure 9, the algorithm took *1327 ms* for complete evaluation compared with *197 ms* for the Gaussian smoothing and *795 ms* for the *Phase-I* stage on an Intel core-i7 4820K processor, with 64 GB RAM. The average time taken by the proposed method for a single brain section of $18K \times 24K$ pixels on an Intel core-i7 4820K processor based CPU with 64 GB RAM is $4-5$ *mins*. Despite having doubled the number of training samples, the competing supervised methods did not exhibit better performance, indicating that evaluation of the model is sound. A 5-fold study has been performed for shallow ML models (5 different combinations of randomly selected sample sets to form the training and test partitions). For deep-CNN we have done a 2-fold study; and our unsupervised method does not require training samples; all samples are used for testing, a few are used to empirically obtain the optimal parameter values.

Regions of Brain	**Proposed**		**SVM** [13]		**RCNN** [43], [19]	
	Pr	**Rcl**	**Pr**	**Rcl**	**Pr**	**Rcl**
Strong Overlap	**0.946**	**0.975**	0.763	0.703	0.649	0.754
Minimal overlap	**0.955**	**0.980**	0.897	0.862	0.799	0.826

TABLE III: Comparative Performance over the strong and minimal overlap regions of cells (from $\simeq 500$ scans of two mouse brains), for the proposed, SVM and RCNN methods.

VI. Discussion

Our approach allows for robust and efficient estimation of the centers of GFP tagged cell nuclei with relatively high precision and recall. Our unsupervised and iterative algorithm based on Ridges and distance transform [37] combined with arc-based region filling provides the best precision value amongst the different approaches used to solve the detection problem. It is important to note that all the methods (including shallow and deep learning methods) also perform well in cases where there is marginal or no overlap (sparse distribution) between the GFP tagged interneurons. Our algorithm approaches near ideal performances in these cases, with the precision-recall values $\simeq$ 0.98 and 1 respectively. The cases where the other models fail are the complex case of strong overlap of cells, where our method gave the best result with a overall precision of 0.972 and recall of 0.995. The drawback with our algorithm is the relatively longer evaluation time frame which is directly proportional to the complexity of the image to be evaluated. Our proposed algorithm is completely automated and unsupervised in its application. We have applied this algorithm to analyze whole brains and a web implementation of the iterative algorithm has been implemented at the mouse brain architecture portal at the Cold Spring Harbor Lab (CSHL) [1] for analyzing cell type data.

It was surprising to note that standard supervised classifiers, e.g. SVM, ADABOOST, etc. and state-of-the-art DL based detection techniques (RCNN and SegNET) did not perform satisfactorily when compared to our method. We believe that this may be because of our method of treating strong overlap decomposition explicitly which improves our performance to $\sim 98\%$, the precision levels needed for real world application. The performance of the ML models using our method is comparable with a previous study on brains [23] where similar methodology of GFP tagging in mice were used and the detection was performed with CNN in order to achieve 90% performance. The difficulties for the network may have been due to the degree of overlap of objects in the image, which, in turn, depends on cell densities as well as microscopic methodology.

VII. Conclusion

We have developed an iterative approach (unsupervised) for efficiently detecting GFP tagged cell centers of GABAergic neurons in the whole mouse brain,. Our results demonstrate that our algorithm, which is based on classical machine vision methods substantially outperform published CNN-based methods. The detection problem that we have solved is of practical interest to neuroscientists. The classical Machine Vision based method we have designed, directly solves the overlap decomposition problem with high precision and recall values. Recent methods of supervised object detection do not provide the level of performance required particularly in cases of strong overlap of cells, as the foreground blobs appear with a wide variety of non-unique two-dimensional patterns that are of different shapes, sizes and structures (due to varied spatial arrangements of cells in strong overlap), making the training data unreliable for efficient learning. Our unsupervised approach provides a much superior performance than recent supervised object detection algorithms such as ADABOOST [48], SVM [13], [34], SegNet [4] and RCNN [19]. Previously published results [23] on similar problems only reach 90% performance level (in contrast with $> 98\%$ of ours). The failure of both the deep and shallow nets to reach $> 90\%$ accuracy levels, is mainly attributable to the difficulties of accurately detecting overlapped nuclei. While our method may not generalize to other unrelated problems, it addresses an important problem in the growing field of computational neuroanatomy of whole brains. CNNs and SVMS are very powerful ML architectures but need further refinements to solve every real-world problem. In our use case the structure of the problem demands a different architecture.

With an ever increasing dataset and information about the neurons and their connectivity, it becomes important to develop automated unsupervised algorithms for efficient data analysis of relevant information [3]. We have provided one approach towards this goal which helps neuroscientists to achieve few targets with greater confidence.

VIII. Acknowledgment

The work was supported by the HN Mahabala chair professorship at Indian Institute of Technology, Madras, The Crick-Clay Professorship at the Cold Spring Harbor Laboratory to Dr. P. Mitra. The authors also acknowledge the support from the Mathers foundation, NIDA grant (5R01DA036400-03) and NSF EAGER grant (1450957).

References

[1] http://mouse.brainarchitecture.org/hua.
[2] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1313–1321, 2016.
[3] R. Armañanzas and G. A. Ascoli. Towards the automatic classification of neurons. *Trends in Neurosciences*, 38(5):307–318, 2015.
[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.
[5] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
[6] J. W. Bohland, C. Wu, H. Barbas, H. Bokil, M. Bota, H. C. Breiter, H. T. Cline, J. C. Doyle, P. J. Freed, R. J. Greenspan, et al. A proposal for a coordinated effort for the determination of brainwide neuroanatomical connectivity in model organisms at a mesoscopic scale. *PLoS Comput Biology*, 5(3):e1000334, 2009.
[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.
[8] I. Castillo, F. J. Kampas, and J. D. Pintér. Solving circle packing problems by global optimization: numerical results and industrial applications. *European Journal of Operational Research*, 191(3):786–802, 2008.
[9] M. Chalfie. Green fluorescent protein as a marker for gene expression. *Trends in Genetics*, 10(5):151, 1994.
[10] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *pacific journal of mathematics*, 181(1):57–88, 1997.
[11] A. Chou. Np-hard triangle packing problems. *Research Science Institute Summer Program for Highschool Students*, 2016.
[12] C. R. Collins and K. Stephenson. A circle packing algorithm. *Computational Geometry*, 25(3):233–256, 2003.
[13] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
[14] H. T. Croft, R. K. Guy, and K. J. Falconer. *Unsolved problems in geometry*. 1991.
[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 248–255. IEEE, 2009.
[16] L. Deng, D. Yu, et al. *Deep learning: methods and applications*, volume 7. Now Publishers, Inc., 2014.
[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
[18] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 130–137. Springer, 1998.
[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.
[20] E. L. Hinrichsen, J. Feder, and T. Jøssang. Random packing of disks in two dimensions. *Physical Review A*, 41(8):4199, 1990.
[21] N. Kasthuri and J. W. Lichtman. The rise of the 'projectome'. *Nature Methods*, 4(4):307–309, 2007.
[22] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *Medical image analysis*, 22(1):77–88, 2015.
[23] Y. Kim, K. U. Venkataraju, K. Pradhan, C. Mende, J. Taranda, S. C. Turaga, I. Arganda-Carreras, L. Ng, M. J. Hawrylycz, K. S. Rockland, H. S. Seung, and P. Osten. Mapping social behavior-induced brain activation at cellular resolution in the mouse. *Cell reports*, 10(2):292–305, 2015.
[24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2556–2563. IEEE, 2011.
[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
[26] A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, 2002.
[27] C. O. López and J. E. Beasley. A heuristic for the circle packing problem with a variety of containers. *European Journal of Operational Research*, 214(3):512–525, 2011.
[28] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 2929–2936. IEEE, 2009.
[29] T. Martinez, L. Vitorino, F. Fages, and A. Aggoun. On solving mixed shapes packing problems by continuous optimization with the cma evolution strategy. In *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on*, pages 515–521. IEEE, 2013.
[30] C. R. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003.
[31] P. P. Mitra. The circuit architecture of whole brains at the mesoscopic scale. *Neuron*, 83(6):1273–1283, 2014.
[32] F. Mualla, S. Schöll, B. Sommerfeldt, A. Maier, and J. Hornegger. Automatic cell detection in bright-field microscope images using SIFT, random forests, and hierarchical clustering. *IEEE Transactions on Medical Imaging*, 32(12):2274–2286, 2013.
[33] S. C. Neoh, W. Srisukkham, L. Zhang, S. Todryk, B. Greystoke, C. P. Lim, M. A. Hossain, and N. Aslam. An intelligent decision support system for leukaemia diagnosis using microscopic blood images. *Scientific reports*, 5, 2015.
[34] S. C. Neoh, W. Srisukkham, L. Zhang, S. Todryk, B. Greystoke, C. P. Lim, M. A. Hossain, and N. Aslam. An intelligent decision support system for leukaemia diagnosis using microscopic blood images. *Scientific reports*, 5, 2015.
[35] D. L. Olson and D. Delen. *Advanced data mining techniques*. Springer Science & Business Media, 2008.
[36] P. Osten and T. W. Margrie. Mapping brain circuitry with a light microscope. *Nature methods*, 10(6):515–523, 2013.
[37] D. W. Paglieroni, G. E. Ford, and E. M. Tsujimoto. The position-orientation masking approach to parametric search for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):740–747, 1994.
[38] V. Pinskiy, J. Jones, A. S. Tolpygo, N. Franciotti, K. Weber, and P. P. Mitra. High-throughput method of whole-brain sectioning, using the tape-transfer technique. *PloS one*, 10(7):e0102363, 2015.
[39] P. Quelhas, M. Marcuzzo, A. M. Mendonça, and A. Campilho. Cell nuclei and cytoplasm joint segmentation using the sliding band filter. *IEEE Transactions on Medical Imaging*, 29(8):1463–1473, 2010.
[40] C. Rafael Gonzalez and R. Woods. Digital image processing. *Pearson Education*, 2002.
[41] H. Rahmani. Packing degree optimization of arbitrary circle arrangements by genetic algorithm. *Granular Matter*, 16(5):751–760, 2014.
[42] N. K. Ratha, S. Chen, and A. K. Jain. Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, 1995.
[43] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.
[44] O. Sporns. The human connectome: a complex network. *Annals of the New York Academy of Sciences*, 1224(1):109–125, 2011.
[45] A. Suissa-Peleg, D. Haehn, S. Knowles-Barley, V. Kaynig, T. R. Jones, A. Wilson, R. Schalek, J. W. Lichtman, and H. Pfister. Automatic neural reconstruction from petavoxel of electron microscopy data. In *Microscopy and Microanalysis*, volume 22, pages 536–537. Cambridge Univ Press, 28 July 2016.
[46] H. Taniguchi, M. He, P. Wu, S. Kim, R. Paik, K. Sugino, D. Kvitsani, Y. Fu, J. Lu, Y. Lin, et al. A resource of cre driver lines for genetic targeting of gabaergic neurons in cerebral cortex. *Neuron*, 71(6):995–1013, 2011.

[47] Y. Toyoshima, T. Tokunaga, O. Hirose, M. Kanamori, T. Teramoto, M. S. Jang, S. Kuge, T. Ishihara, R. Yoshida, and Y. Iino. Accurate automatic detection of densely distributed cell nuclei in 3d space. *PLoS Comput Biol*, 12(6):e1004970, 2016.

[48] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2001.

[49] H. Wang, W. Huang, Q. Zhang, and D. Xu. An improved algorithm for the packing of unequal circles within a larger containing circle. *European Journal of Operational Research*, 141(2):440–453, 2002.

Supplementary Document for manuscript entitled: High precision automated detection of labeled nuclei in terabyte-scale whole-brain volumetric image data of Mouse

Girraj Pahariya*, Sukhendu Das*, Jaikishan Jayakumar*, Samik Banerjee*,
Venu Vangala*, Keerthi Ram†, Partha Mitra‡
*Computer Science & Engg. Dept., IIT Madras, Chennai, IN 600036
†Healthcare Technology Innovation Centre, IIT Madras, Chennai, IN 600036
‡Cold Spring Harbor Laboratory, NY 11724

S.1. Intermediate results of the different steps in the sub-processes: *Phase-I* and *Phase-II*

This supplementary section shows the intermediate results of both the sub-processes in the proposed pipeline (figure 4) for cell detection. It also illustrates the optimal parameter values used in the algorithm as a table. Figure S.1 illustrates the results of the intermediate steps in *Phase-I* sub-process, for the example given in figure 3(d) in main manuscript. Following the steps given in section III of manuscript, results of eight intermediate steps are shown.

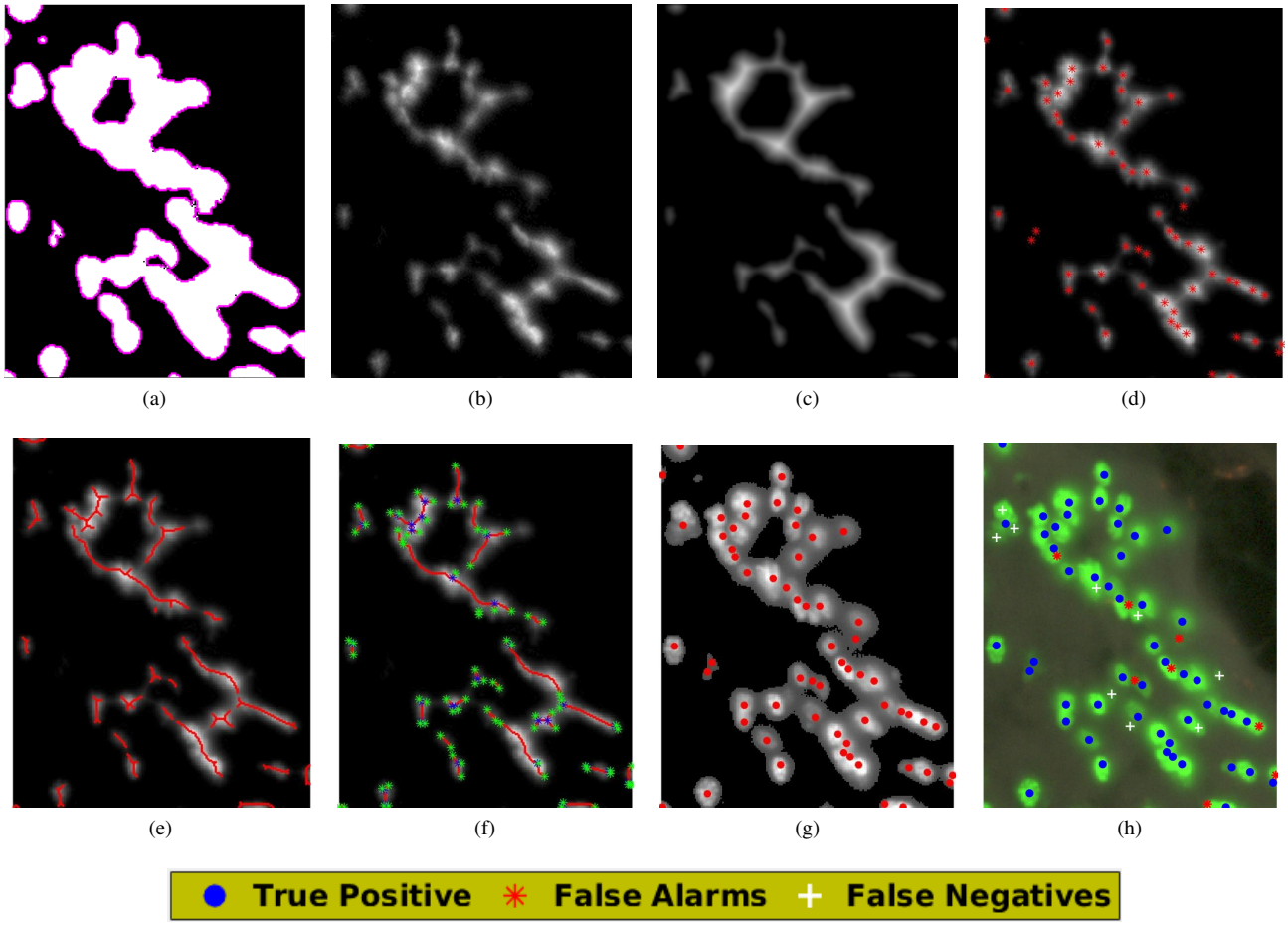


Fig. S.1: (a) Binarized smoothed image (of the example given in in figure 5(a) from main manuscript) overlayed with the edge map obtained using Canny edge detection algorithm; (b) Distance Transform (DT) on (a) over FGL given in figure 3(d) of main manuscript; (c) Modulated DT (mDT) map, obtained by multiplying FGL with DT map (b); (d) Intensity peaks (red dots) marked on mDT map given in (c). (e) the Ridge lines extracted from (c)using the "vessel" filter [7] algorithm; (f) Landmarks such as Bifurcation points (BF) and Ridge Endings (RE) marked on Ridge lines in (e), which correspond to potential centers of detected cell nuclei; (g) Final set of Cell Centers obtained by merging (union and clustering) nearby cell center positions obtained in (d) and (f); (h) Comparative evaluation of *Phase-I* output compared with the ground truth showing the true positives, false alarms and false negatives marked on enhanced image in figure 3(d) (of main manuscript).

The figure S.2 shows the results of the intermediate steps in *Phase-II* sub-process, for the example given in figure 3(d) of the main manuscript. Following the steps given in section IV of manuscript, results of eight intermediate steps are shown.

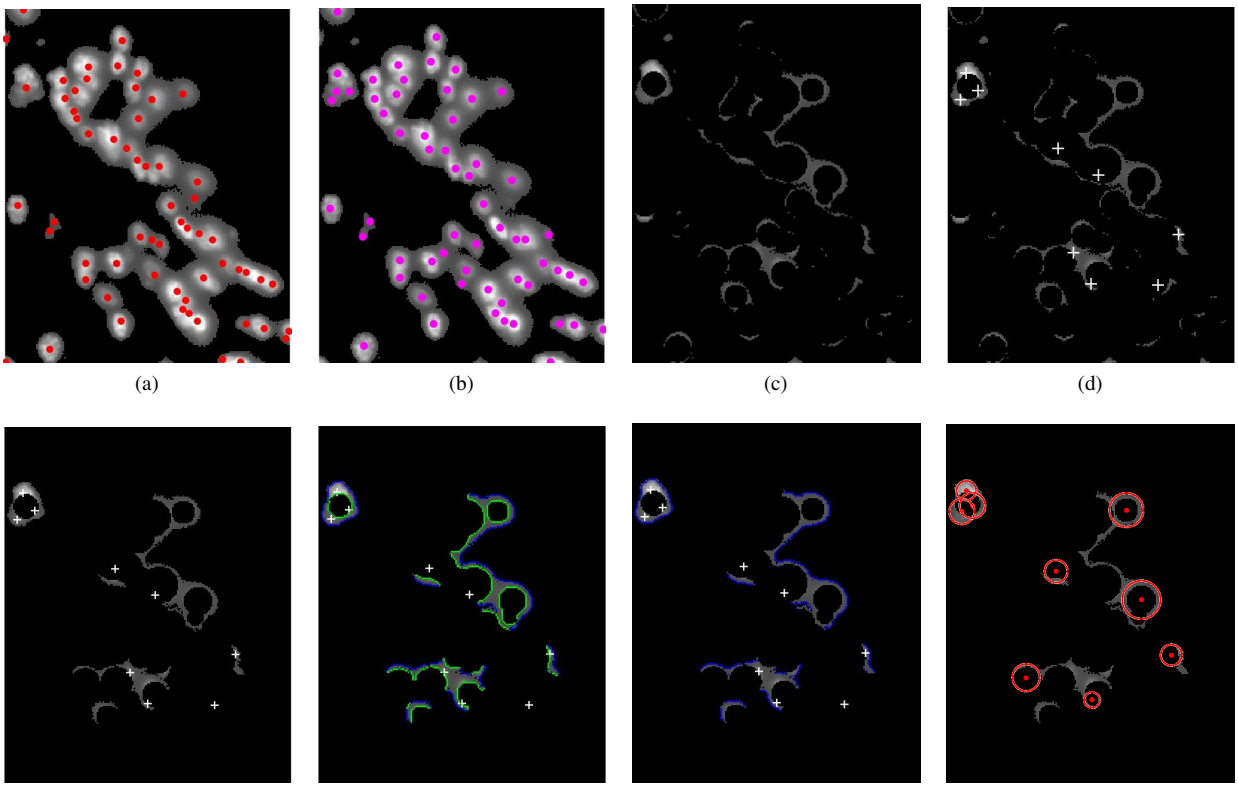


Fig. S.2: Illustration of the results of the intermediate steps in Phase-II sub-processing stage (see figure 6 in main manuscript) for the example of strong overlap of GFP nuclei. (a) Centers detected by Phase-I of processing on FGL, as given in S.1(g); which serves as the input of Phase II (b) Manually annotated Ground Truth superimposed on the FGL (see figure 3(d) of manuscript); (c) RFM (Residual foreground map) obtained after the first iteration, from (a); (d) Cells undetected by the Phase-I stage of processing, shown superimposed on RFM marked using white crosses; (e) undetected cells, on significant part of RFM extracted from (d); (f) IBC in green and OBC in blue on edges of RFM in (e); (g) filtered (prominent) convex OBCs with undetected set of cells marked in red; and (h) cell centers detected by fitting circles using Hough Transform on convex OBCs in (g). Figure best viewed in color.

A. Definitions of parameters used in the manuscript

True Positives: *denote the cell-centers that are detected correctly by our algorithm and are present in the ground truth.*
False Alarms: *denote the cell-centers that were missed by our algorithmbut were present in the ground truth.*
False Negatives: *denote the cell-centers that were detected by our algorithm but were not present in the ground truth.*

TABLE S.1: Values of heuristic parameters used in the proposed algorithm (refer sections III & IV of the main manuscript). All values are obtained empirically to be optimal.

Params	Phase	Details & Remarks	Value
TH	I	Threshold, used for segmentation of foreground of green channel	4000
L	I	Minimum Distance between landmark features on Ridge Lines (DT based method)	20 pixels
dc	I	Minimal distance for Merging Centers of cells	12 pixels
CD	II	Average Diameter of cells (exact value obtained using DT)	$4-7$ pixels
A	II	Area threshold for detecting significant connected components in RFM; Phase II - Arc Based Iteration stage	30 pixels
DR	I & II	Evaluation Process	5 pixels
W	-	Window size for NMS (suppression of non-maximas based on confidence values) of RCNN, SVM & ADABOOST methods	11 pixels

S.2. Description of Phase-II sub-process using a synthetic example

The illustration of the iterative process (*Phase-II*) to detect strong overlap of cells, using the synthetic example as shown in Figure 7 of the main manuscript, is as follows:

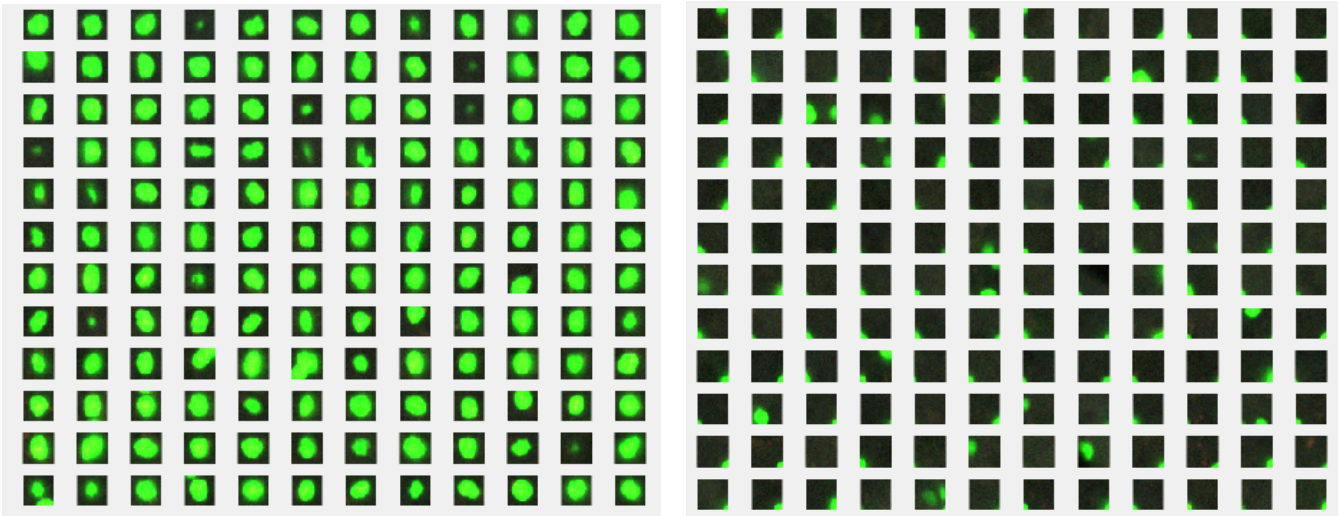
Input: FGL, DT-map of FGL and list of candidate cell centers (obtained at *Phase I*).

- *Phase-I* estimates the edge map of the FGL (shown as the shaded area in figure 7(b)) and a peak estimated in mDT (figure 7(c)) is marked as a red dot.
- This detected cell (as peak in mDT) is preserved and a circle representing that cell is fitted (radius derived from mDT value, with an upper-bound) using the detected peak, as shown in figure 7(c).
- **START LOOP:** The detected cell region is suppressed using the background color and a residual foreground map (RFM) map is obtained, as shown by the shaded area in figure 7(d). Note that, the visual example in figure 7 is a simple synthetic case, used solely for illustration/explanation, and hence devoid of any such small patches to be eliminated.
  1) Edges are detected on the residual map (RFM) and segregated into two parts:
     - *Inner boundary contour (IBC) of detected cells* - shown by a green color circle in figure 7(d); and
     - *Outer Boundary Contours (OBC)* - shown in blue in figure 7(d).
  2) IBC is removed. Three circular convex arcs are detected on the OBC, using zero crossing curvature (ZCC) [11].
  3) Hough Transform [12] is used to fit three circular blobs on the convex segments (arcs) on OBC to fit three new cells.
  4) Centers of these circles fitted (using the Hough Transform [12] process) on the prominent OBCs are considered as the new cell centers (figure 7(f)) and is updated to the list.
  5) Suppress cell area in RFM using background color.
  6) With no significant area left in RFM, terminate and break the loop; else continue.
- **END LOOP:** This process repeats till no uncovered significant connected components in residual layer (RFM) are left to process; else go back to step (1) to iterate the process.
- The final list of new cell centers is merged with the earlier list of cell centers detected from *Phase I*: DT + Ridge based method; final set is shown in figure 7(g).

S.3. Details of Training the supervised ML and DL models used for cell detection, as competing methods

A. Support Vector Machines (SVM)

For training the shallow supervised methods of classification, we have used annotated cell centers from the ground truth. Using this information, template windows of size 25×25 pixels were created over the marked cell bodies. Each window that contains an entire cell body and were used as the positive samples for training. We also used other windows from the remaining parts of the brain that contain only partial or no marked cell bodies within the template window and were used as negative samples. Care was taken so as to avoid those windows whose centers are nearby to the labeled cell centers.. Examples of positive and negative samples used in the SVM are shown in figure S.3(a) and S.3(b) respectively. The training samples were randomly selected from 3 classified regions as: sparsely distributed, minimal and strong overlap of cells in different parts of brain regions based on density of occurrence of the GFP tagged nuclei.

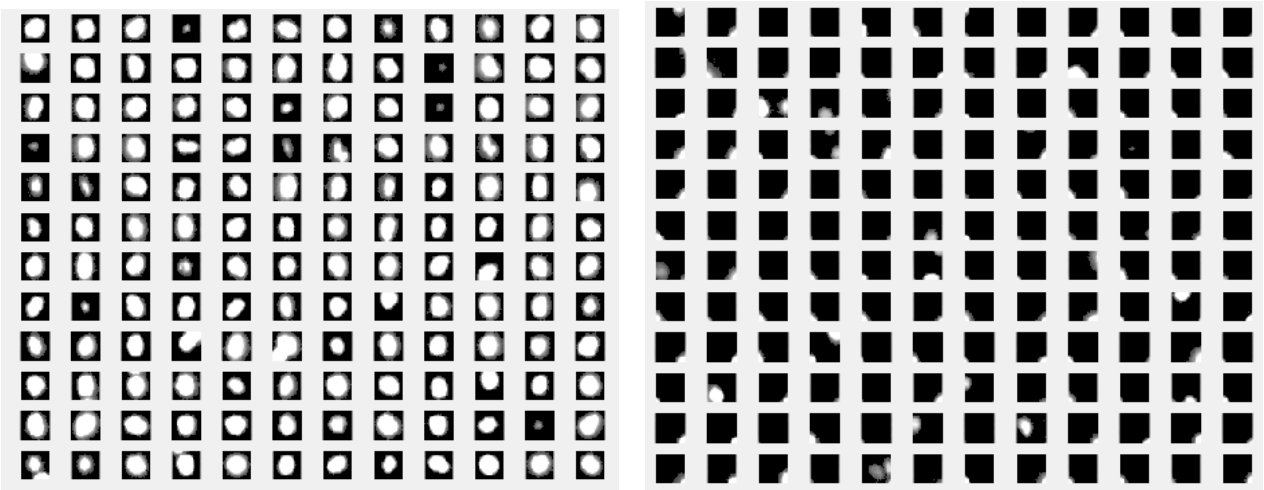


(a) (b)

Fig. S.3: (a) shows the samples consisting of template windows with complete cell nuclei (positive samples) and (b) shows examples with no complete cell nuclei (negative samples) that were used as the the training sets for the SVM [4].

B. ADABOOST (with HAAR features)

For ADABOOST, we have used features that were extracted from the the mDT map and were used to train the the model. Figure S.4 shows examples of negative and positive samples used for training the ADABOOST model. It is important to note the higher fluorescent irradiance at the center of cells in samples of figure S.4, compared to that in figure S.3. Features, such as, Histogram of Oriented Gradients (HOG) [5] and Multi-Texton Histogram (MTH) [10] are extracted from mDT map (equation 1 in manuscript), for each of the negative and positive mDT samples. SVM [4] with Gaussian kernel was used for training.



(a) (b)

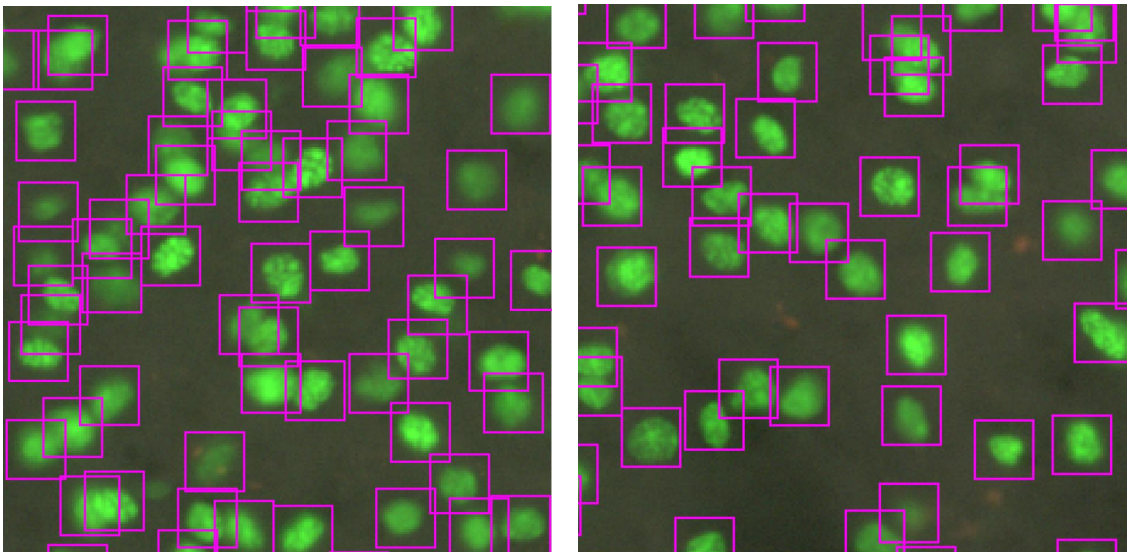
Fig. S.4: Template samples of mDT obtained from those in figure S.3. These are again used for extracting features, required for training the ADABOOST (with HAAR feature) classifier; (a) +*ve* samples, (b) -*ve* samples.

A total of $25K$ positive samples and $50K$ negative samples were used for training. For training the ADABOOST [16] classifier, Haar features are extracted from the samples. The SVM and ADABOOST implementations of MATLAB are taken off-the-shelf. Positive classification is considered as an area of a detected cell and centers of positive windows were considered as detected cell centers.

C. CNN models

Two methods of supervised CNN models, RCNN and SegNet, were also used for experimentation. RCNN [8] performs a bounding box regression on the objects in an image for detection. We have used equal number of samples for training and testing the RCNN model. The training stage takes the $224x224$ pixel images and the coordinates of the bounding boxes around each cell present in the image. A total of 1,47,800 such images are generated each having 2 or more cells in them. We then combine samples from sparse, minimal and strong overlap regions for training the model. Traditionally, RCNN has been used in literature to detect objects, even when the size varies for each object within an image. Since we have only one object of interest i.e. GFP tagged nuclei in our data, we provide a bounding box of size 25×25 pixels around each cell, derived empirically. RCNN [13] uses the coordinates of the positive samples as the anchor boxes. The negative samples are generated implicitly within the network, i.e. one does not have to provide negative samples to the code as input. RCNN takes care of the scale and translation invariance. A large number of such anchor boxes are given as input to the RCNN during the training process and as a result any variability of the cell configuration is taken care during the training. Post training, RCNN takes an image as input and detects the bounding box around the cell bodies as output. Centroids of the bounding boxes obtained as RCNN output were considered as detected cell centers. A Tensorflow based implementation of RCNN [13] is taken of-the-shelf for our experimentation. Samples used as training the RCNN model is shown in figure S.5. The magenta boxes show the anchor boxes that are fed into the RCNN network during the training. These anchors are drawn based on the manually annotated data of cell-centers. These square boxes are positioned at these cell-centers with a side length of 11 pixels. Faster-RCNN [8], [13] was trained using a Tensorflow based implementation for 70K epochs. We have used $\sim 10^6$ positive training samples for faster RCNN (compared to [8], which reports using 105 samples). The anchor boxes are equivalent to those in PASCAL-VOC challenge; where the objects are simple and different. In our case, each window contains an entire cell body and were labeled as positive samples. The RCNN techniques used for COCO (image classification) do not use more than 10K samples per class. Since the problem at hand is a two-class problem and the training set size was $\simeq 10^6$, we aimed to find a trade-off between the training time and the number of samples. We also found that increasing the samples 3-fold did not significantly increase the performance of the system. Our RCNN parameter count is 2.4×10^6 [13]. Our model was trained for 7×10^6 epochs using the codes provided by the authors [13].

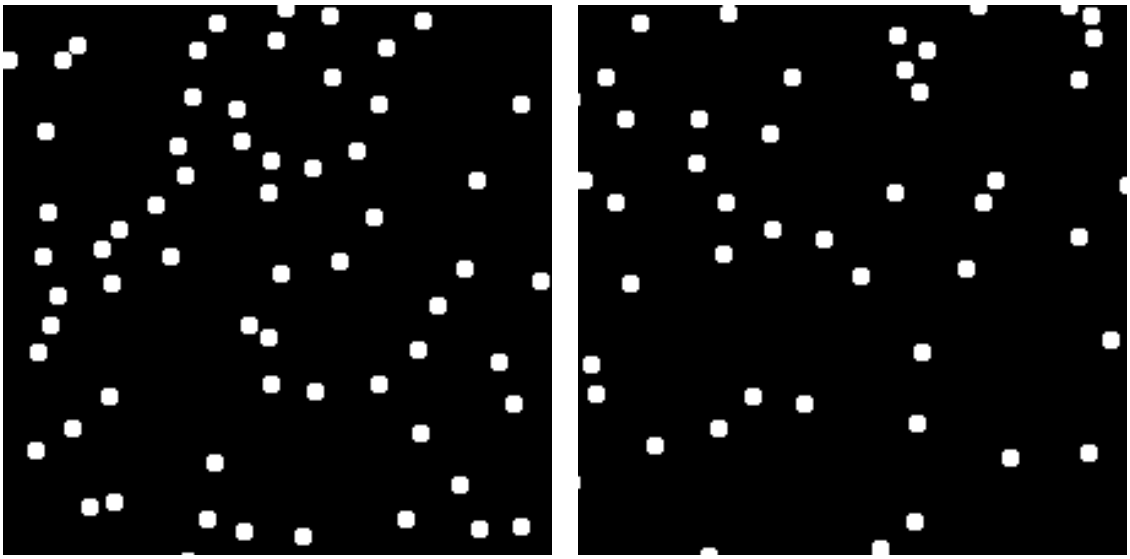
For the problem at hand, recent literature in the domain of Deep Learning does not have an unsupervised framework for CNNs, to the best of our knowledge. Hence, we have resorted to comparing our results with the supervised CNN architectures.



(a) (b)

Fig. S.5: Two sample sections of the brain image used for training the RCNN [8]. Bounding boxes are positioned using human-annotated groundtruth data. No negative samples needed.

The SegNet [2] model has been trained as per the code provided by the authors. It's a supervised (CNN) method used previously for detecting cells in blood. Training samples (positive ones only needed) for SegNet were generated by delineating circular patches around manually annotated cell centers. In this case, we used a circular patch with the radius of 4 pixels around the manually annotated cell-centers, as inputs for training. The trained SegNet map produces outputs as circular patches (sometimes overlapping with each other) on the test image, whose centers are detected as cell-centers. The SegNet [2] is a torch based implementation that is used previously for segmentation in computer vision. The SegNet model is a Variational auto-encoder based model, which inputs RGB image at one end and its binary segmented map on the other end. Based on the annotated cell-centers, circular patches with radius ($r = 4$ pixels) are inpainted to create virtual segmented maps of the original image, for training the SegNet model.



(a) (b)

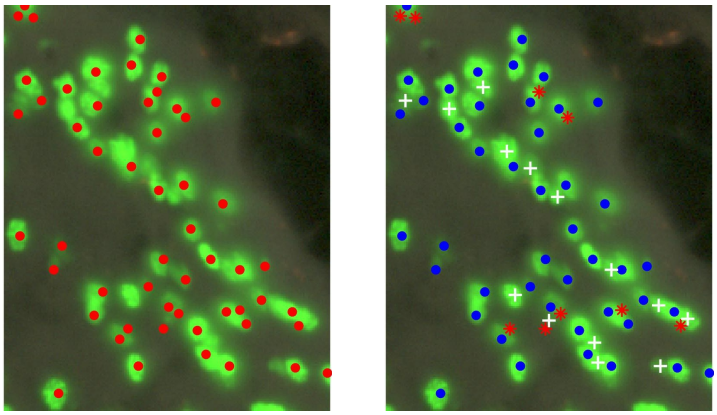
Fig. S.6: Two sample sections of the brain image used for training the SegNet [2]. Circular patches are drawn using human-annotated ground truth data. No negative samples needed.

TABLE S.2: Number of samples used for training and testing (each tile is a cropped section of brain scan of size 512×512 pixels).

Technique	**Training samples**		**Testing Samples**
	+ve	**-ve**	
SVM	$25K$	$50K$	$350K$
Haar + ADABOOST	$25K$	$50K$	$350K$
RCNN	$149K$	-	250 tiles
SegNET	$150K$	-	250 tiles
Proposed	-	-	512 tiles

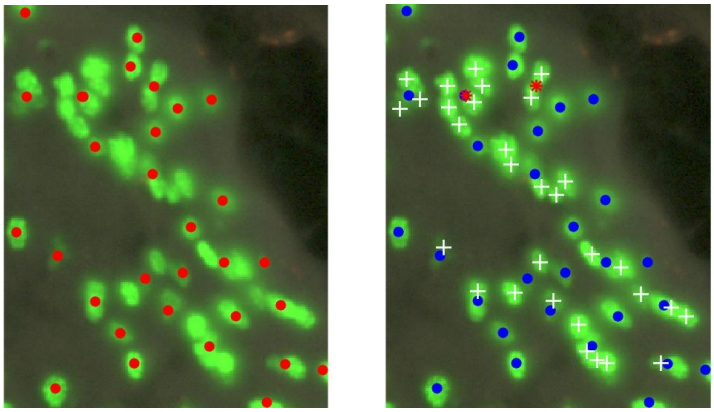
S.4. Analyzing outputs generated by SVM, ADABOOST & DL models (RCNN and SegNet) using NMS

The outputs obtained from the software implementation of RCNN [8], [13] are available as bounding boxes on the test images. The centers are calculated for each of these square boxes to get the cell-centers. A similar strategy is used for the output of SegNet, which produces the segmented maps as circular blobs, where the center for each blob is taken as the detected cell-center. Illustrations are given in figures S.7-S.9 for two cropped regions with strong overlap and one region with minimal overlap of cells in the mouse brain.



Detected cell-centerss7 Detection evaluated with groundtruth

For RCNN

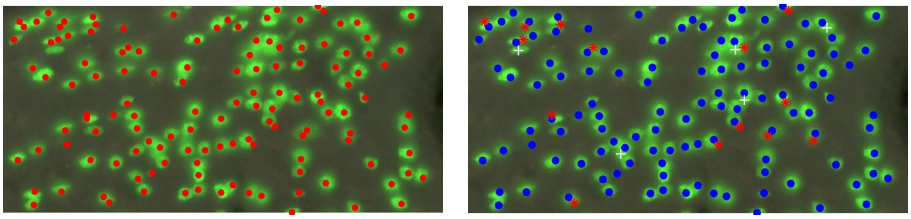


Detected cell-centers Detection evaluated with groundtruth

For SegNet

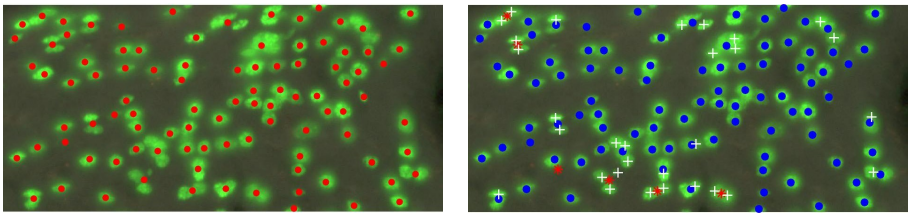
Fig. S.7: **Strong Overlap Region**: Visual illustrations of the results of DL models (RCNN and SegNet) for a region of strongly overlapping cells in mouse brain. From left to right - Output of trained DL model; the detected cell-centers from output; Detection evaluated with groundtruth (best viewed in color). See figure S.1 for interpretation of color markers.

Implementations of RCNN and ADABOOST (using HARR features) classifiers produce multiple bounding boxes, while SVM produces multiple candidate locations as target outputs during the testing phase. These positive responses (output candidates) obtained from the trained machines, typically appear as clusters (of strong overlapping boxes in cases of RCNN and ADABOOST). SVM implementation, on the other hand, produces clusters of positive responses with varying confidence



Detected cell-centers Detection evaluated with groundtruth

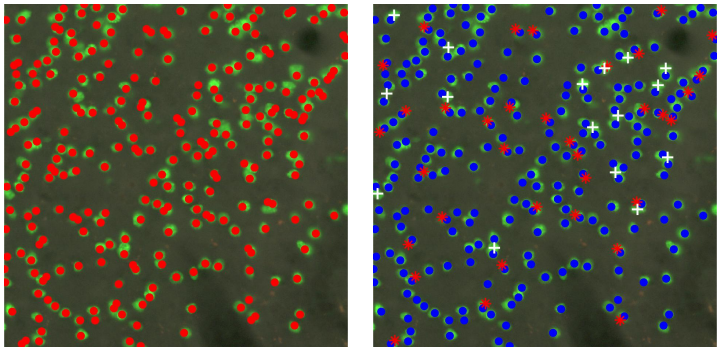
For RCNN



Detected cell-centers Detection evaluated with groundtruth

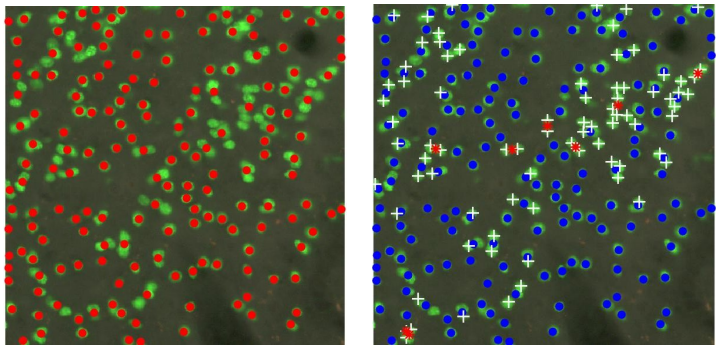
For SegNet

Fig. S.8: **Strong Overlap Region**: Visual illustrations of the results of DL models (RCNN and SegNet) for a region of strongly overlapping cells in mouse brain. From left to right - Output of trained DL model; the detected cell-centers from output; Detection evaluated with groundtruth (best viewed in color). See figure S.1 for interpretation of color markers.



Detected cell-centers Detection evaluated with groundtruth

For RCNN



Detected cell-centers Detection evaluated with groundtruth

For SegNet

Fig. S.9: **Minimal Overlap Region**: Visual illustrations of the results of DL models (RCNN and SegNet) for a region of minimal overlapping set of cells in mouse brain. From left to right - Output of trained DL model; the detected cell-centers from output; Detection evaluated with groundtruth (best viewed in color). See figure S.1 for interpretation of color markers.

levels. The same is true for the other two methods, where each output appears with a confidence score along with a scale of the bounding box.

Most implementations of RCNN, ADABOOST and SVM use a method of post-processing (based on local averaging) on these nearby candidates, to obtain the final target position. These implementations did not produce satisfactory results and have thus been modified by us. The primary reason for this failure is that, in general their design does not deal with our scenario of minimal and strong overlapping presence of multiple identical objects (e.g. thin crowd, swarm of bees, ant colony etc.) as targets [14].

To obtain a better result for finalization of the target (cell) locations, we have used a Non-maximal Suppression (NMS) [15], [3] strategy. The solution demands the detection of a maximal (local) response from a 2-dimensional clutter of close-by responses with varying confidence scores. We have used a minor modification of the post-processing from the implementation of [6], as a greedy method of NMS. The modification involves: Within a large neighborhood, $\boldsymbol{W}$, instead of averaging the overlapping window responses, we consider a rank-ordered list of candidates sorted by the confidence scores. While doing so we eliminate outliers of extremely small and large scales (range learned using responses from GT data). Lastly, choose the topmost confidence scores as winner (max) within the local window ($\boldsymbol{W}$), and update the list to apply this last step of NMS iteratively till there are no further significant peaks of confidence score left as a candidate response. This method works reasonably well for our specific scenario of minimally and strongly overlapping objects (brain cells). The qualitative (visual) results of the supervised processes: SVM, HAAR+ADABOOST and RCNN, along with their evaluation studies given in tables 1-3, are all hence reported using the outputs of this NMS-based process.

S.5. Additional Results and Extended Discussions

This section exhibits a few additional results of different (competing) methods used to get the cell-centers. The following figures shows the output of competing methods (see table S.2) on two overlap and one minimal overlap regions of cells in the mouse brain scan. Results of the proposed method appear in main manuscript (figures 9 and 10).

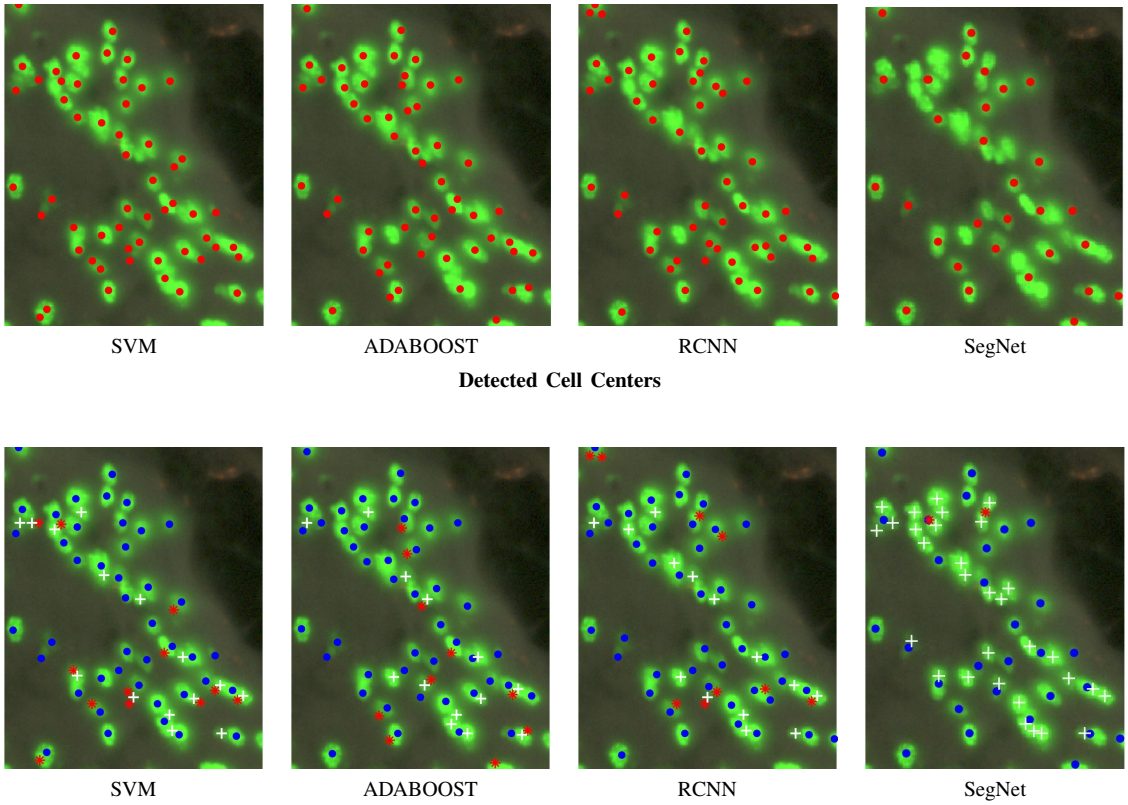


Fig. S.10: **Strong Overlap Region**: Visual illustrations for qualitative comparative study of performance of different competing methods, for the image given in figure 3(d) in main manuscript (best viewed in color). See figure S.1 for interpretation of color markers.

Section II-A in main manuscript gives the basis and justifies the design of our proposed unsupervised pipeline. Other alternative strategies (for *Phase-II*) which we had pondered and partly explored for the solution to the problem of cell detection

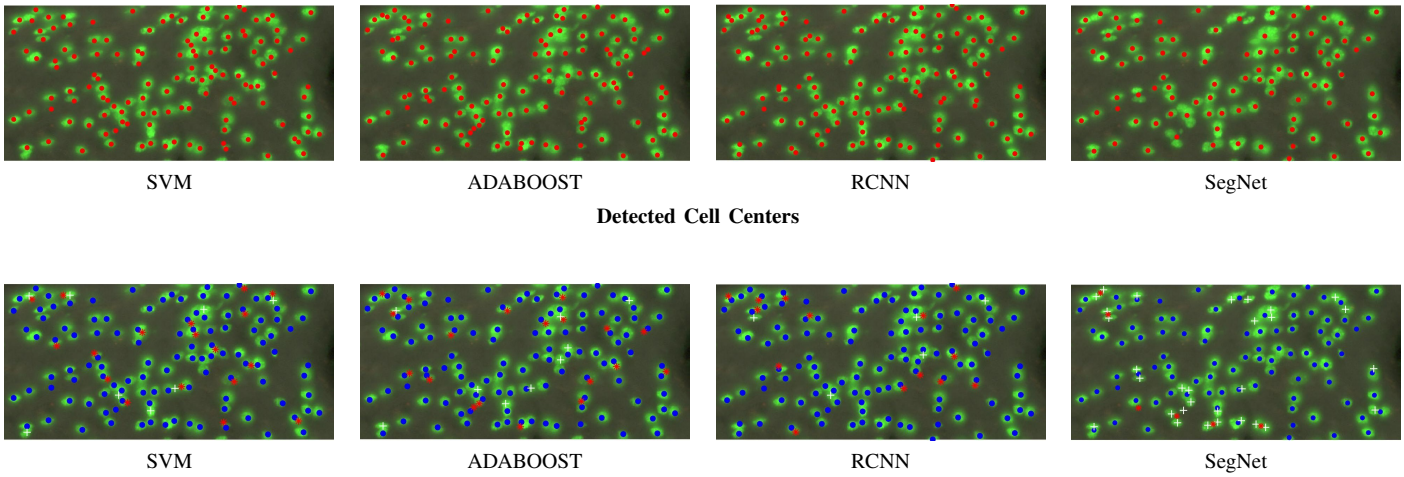


Fig. S.11: **Strong Overlap Region**: Visual illustration for qualitative comparative study of performance of different competing methods, for the image given in figure 3(d) in main manuscript (best viewed in color). See figure S.1 for interpretation of color markers.

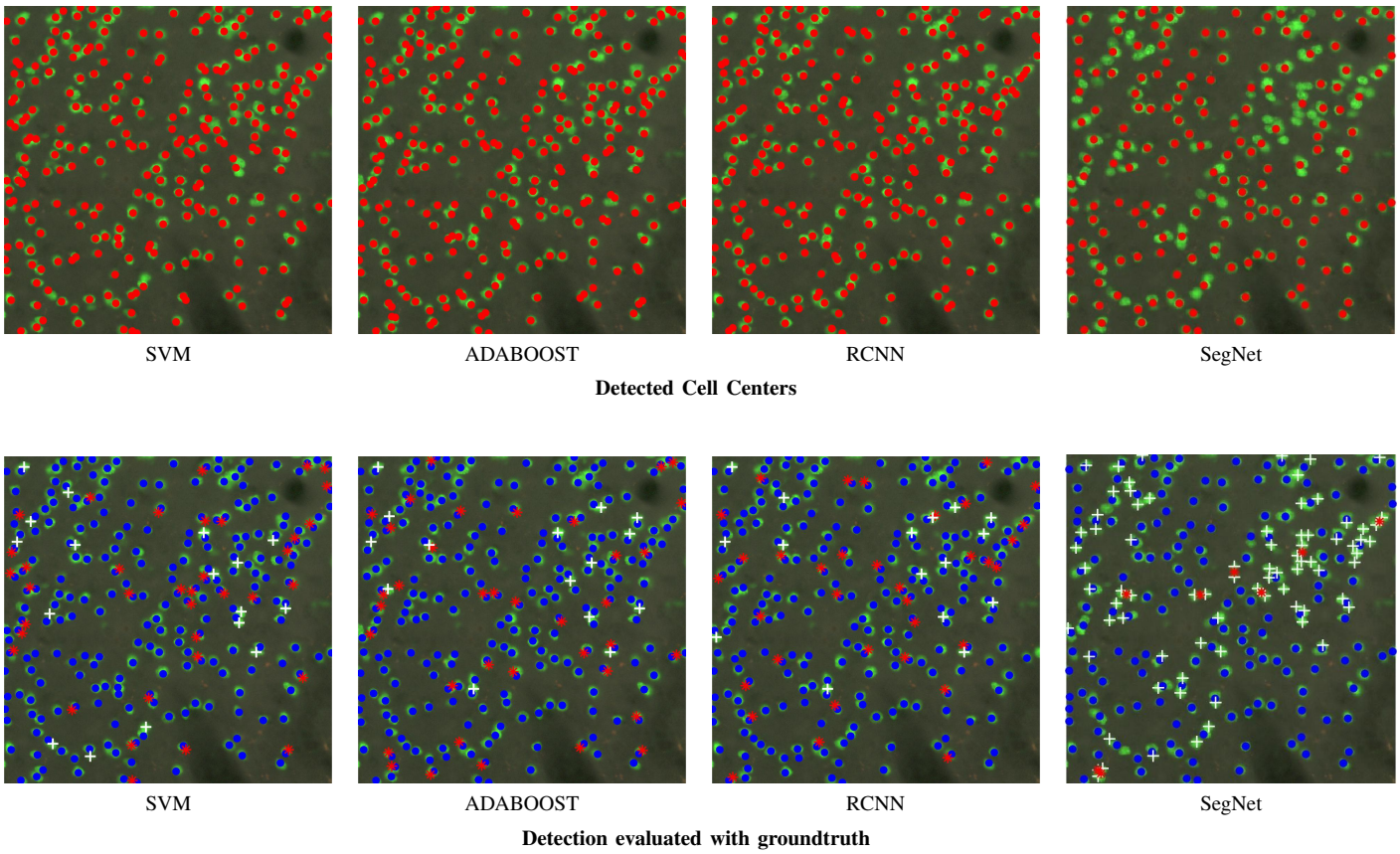


Fig. S.12: **Minimal Overlap Region**: Visual illustration for qualitative comparative study of performance of different competing methods, for the image given in figure 10(b) in main manuscript (best viewed in color). See figure S.1 for interpretation of color markers.

are: (i) 2-D bin packing problem; (ii) Minimize an optimization function for placement of cells, with suitable scores and cost criteria; and (iii) image saliency based on superpixels [1]. A few trials did not give us encouraging results (*e.g.*, trapped in a false minima while minimizing), which were worse than even those produced by the simplest Gaussian convolution (see table 1, section VI-A in manuscript) operation. We are certain that this dataset will throw theoretical and analytical challenges in overlapping fields of image processing (and CV), computational geometry and variational optimization.

One may have an impression that the inter-neuron somata are easier to detect than pyramidal cell somata. This is not necessarily true: the difficulties here have to do with the degree of overlap of objects in the image, which depends on cell densities as well as microscopic methodology. The previous results, published in [9], only reach 90% performance level (in contrast with our performance levels of $> 98\%$), something that is easily seen on visual examination of whole-brain images from [9] (not shown here). The proposed algorithm is completely automated and unsupervised in its application.

S.6. Samples of GUI interface in the tool used for manual annotation

Figures S.13 and S.14 show the screenshots of the GUI for the manual annotation tool, used to obtain ground-truth.

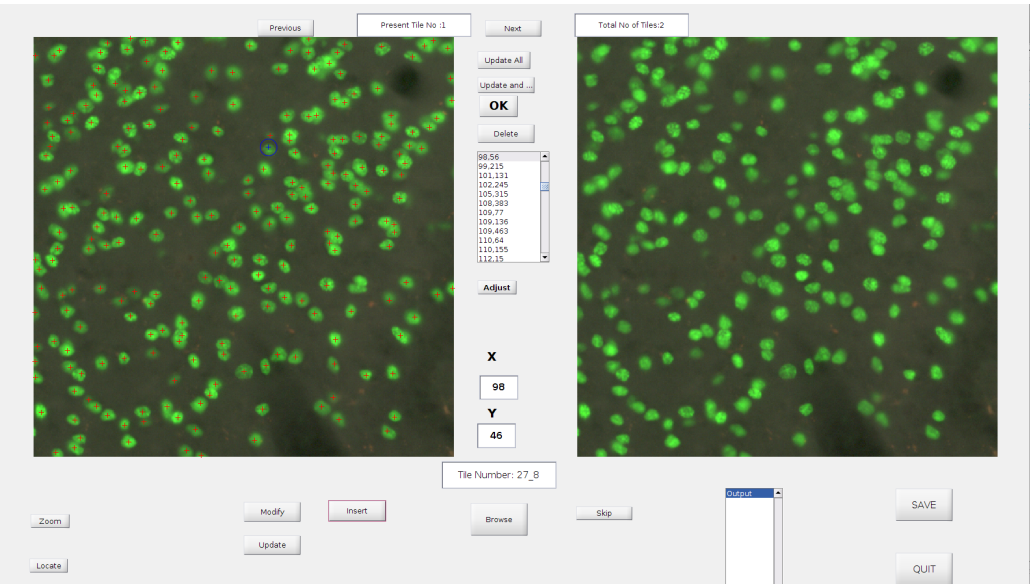


Fig. S.13: Illustration of the manual annotation tool over a minimal overlap region of the brain scan.

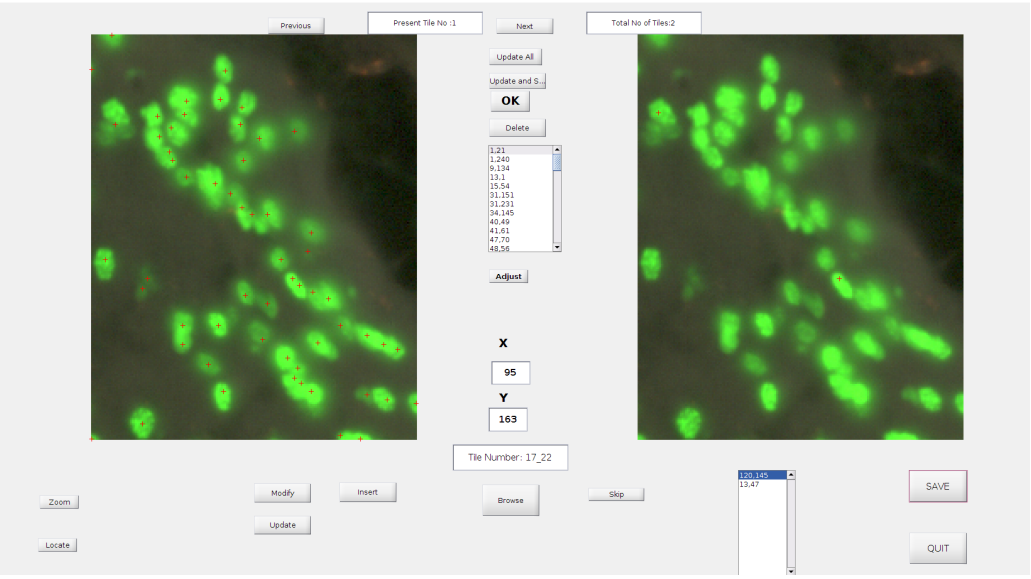


Fig. S.14: Illustration of the manual annotation tool over a region of cell overlap in the brain scan.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 510–517. IEEE, 2005.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 130–137. Springer, 1998.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.

[9] Y. Kim, K. U. Venkataraju, K. Pradhan, C. Mende, J. Taranda, S. C. Turaga, I. Arganda-Carreras, L. Ng, M. J. Hawrylycz, K. S. Rockland, H. S. Seung, and P. Osten. Mapping social behavior-induced brain activation at cellular resolution in the mouse. *Cell reports*, 10(2):292–305, 2015.

[10] G.-H. Liu, L. Zhang, Y.-K. Hou, Z.-Y. Li, and J.-Y. Yang. Image retrieval based on multi-texton histogram. *Pattern Recognition*, 43(7):2380–2389, 2010.

[11] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381, 1998.

[12] C. Rafael Gonzalez and R. Woods. Digital image processing. *Pearson Education*, 2002.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[14] R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian Conference on Computer Vision*, pages 290–306. Springer, 2014.

[15] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2001.