AUTOMATED COUNTING OF LABELLED CELLS IN RODENT BRAIN SECTION IMAGES

F.J. Theis\textsuperscript{1}, Z. Kohl\textsuperscript{2}, H.G. Kuhn\textsuperscript{2}, H.G. Stockmeier\textsuperscript{1} and E.W. Lang\textsuperscript{1}

\textsuperscript{1}Institute of Biophysics, University of Regensburg, 93040 Regensburg, Germany
\textsuperscript{2}Department of Neurology, University of Regensburg, 93053 Regensburg, Germany
email: fabian@theis.name

ABSTRACT
The genesis of new cells, especially of neurons, in the adult human brain is currently of great scientific interest. In order to measure neurogenesis in animals new born cells are labelled with specific markers such as BrdU; in brain sections these can later be analyzed and counted through the microscope. So far, the image analysis has been performed by hand. In this work, we present an algorithm to automatically segment the digital brain section picture into cell and noncell components, giving a count of the number of cells in the section. This is done by first training a so-called cell classifier with cell and non-cell patches in a supervised manner. This cell classifier can later be used in an arbitrary number of sections by scanning the section and choosing maxima of this classifier as cell center locations. For training, single- and multi-layer perceptrons were used. In preliminary experiments, we get good performance of the classifier.

KEY WORDS
Cell counting, image segmentation, cell classification, neurogenesis, BrdU

1 Biological background

1.1 New neurons in the adult brain

During the last decades the fact that new neurons are continuously generated in the adult mammalian brain - a phenomenon termed \textit{adult neurogenesis} - came more and more into focus of neuroscience research [1][2][7]. Under physiological conditions, neuroscientists found, that adult neurogenesis seems to be restricted to two brain regions: The wall of the lateral ventricle and the granular cell layer of the hippocampus.

A large variety of factors including environmental signals, trophic factors, hormone and neurotransmitters have recently been identified to regulate the generation of new neurons in the adult brain. These studies were typically performed by using a combination of different histological techniques, such as non-radioactive labeling of newly generated cells, stereological counting and confocal microscope analysis, in order to quantitatively analyze adult neurogenesis (review in [8]). However, this procedure is time consuming, since histological analysis currently depends on assessment of positive signals in histological sections by individual investigators through manual or semiautomatic counting.

1.2 Method used

\textit{Bromodeoxyuridine (BrdU)}, a thymidine analog is given systemically and is integrated into the replicating DNA during cell division [3]. Using a specific antibody against BrdU, labelled cells can be detected by an immunohistochemical staining procedure. The nuclei of labelled cells on 40\mu m thick brain sections appear in dark brown or black dense color. To determine the amount of BrdU-positive cells in the granular cell layer of the hippocampus they were counted on a light microscope (Olympus IX 70; Hamburg, Germany) with a 20\times objective. Digital images with a resolution of 1600 \times 1200 pixels were taken by a color video camera using the analySIS-software system (Soft Imaging System, Münster, Germany).

2 Automated counting

Figure 1 shows a section image, in which the cells are to be counted.

Classical approaches such as thresholding and erosion after image normalization were not successful, mainly because cell clusters in the image cannot be detected properly and counted using this method.

We decided to adapt a method proposed by Nattkemper \textit{et al.} [9] to evaluate fluorescence micrographs of lymphocytes invading human tissue. The main idea is to build in a first step a function mapping an image patch to a \textit{confidence value} in [0, 1], indicating how probable a cell lies in this patch or not — we call this function \textit{cell classifier}. In the second step this function is used as a local filter on the whole image; its application gives a probability distribution over the whole image with local maxima at cell positions. Nattkemper \textit{et al.} call this distribution \textit{confidence map}. Maxima analysis of the confidence map reveals the number and the position of the cells (image segmentation).

3 Cell classifier

In this section, we will explain how to generate a cell classifier that is a function mapping image patches to cell confidence values. For this we will generate a sample set of cells...
3.1 Sample set

After fixing the patch size — in the following we will use 20 by 20 pixel grey-level image patches — a training set of cell and non-cell patches has to be generated manually by the expert. These image patches are then to be classified by a neural network. Figure 2 shows some cell and non-cell patches.

Interpreting each 20 by 20 image patch as a 400-dimensional vector, we thus get a set of \( L \) training vectors

\[
T := \{(x_1, t_1), \ldots, (x_L, t_L)\}
\]

with \( x_i \in \mathbb{R}^n \) — here \( n = 20^2 \) — representing the image patch and \( t_i \in \{0,1\} \) either 0 or 1 depending on whether \( x_i \) is a non-cell or a cell. The goal is to find a mapping correctly classifying this data set that is a mapping \( \zeta : \mathbb{R}^n \to [0,1] \) with \( \zeta(x_i) = o_i \) for \( i = 1, \ldots, L \). We call such a mapping cell classifier. Of course \( \zeta \) is not uniquely defined by the above property, so some regularization has to be introduced. Any interpolation technique such as Fourier or Taylor approximation can be used to find \( \zeta \); we will use single and multilayer perceptrons as explained in the previous section.

3.2 Preprocessing

Before we apply neural network learning, we preprocess the data as follows: After mean removal, we apply principal component analysis (PCA) in order to reduce the data set dimension as well as to decorrelate the data in a first separation step. This is achieved by diagonalizing the data set covariance and projecting along the eigenvectors with largest eigenvalues.

By only taking the first 5 eigenvalues of the training set, projection along those first 5 principal axes still retains 95% of the data. Thus, the 400-dimensional data space was reduced to a whitened 5-dimensional data set.

A visualization of the 120-samples data set is given in figure 3, here after projection to 3 dimensions. One can easily see that the cell and non-cell components can be linearly separated — thus using a perceptron, see next section, can indeed already learn the cell classifier. Furthermore, a k-means clustering algorithm has been applied with \( k = 2 \) in order to find the two data clusters. Those directly correspond to the cell/non-cell components, see figure.

3.3 Neural network learning

Supervised learning algorithms try to approximate a given function \( f : \mathbb{R}^n \to A \subset \mathbb{R}^m \) by using a number of given sample-observation pairs \( (x_\lambda, f(x_\lambda)) \in \mathbb{R}^n \times A \). We will restrict ourselves to feed-forward layered neural networks [4]; furthermore, we found that simple single-layered neural networks (perceptrons) in comparison to multi-layered networks (MLP) already sufficed to learn the data set well — furthermore they have the advantage of easier rule extraction and interpretation.
A perceptron with output dimension 1 consists of a single neuron only, so the output function $y$ can be written as

$$y(x) = \theta(w^T x + w_0)$$

with weight $w \in \mathbb{R}^n$, $n$ input dimension, $w_0 \in \mathbb{R}$ the bias and as activation function $\theta$, the Heaviside function ($\theta(x) = 0$ for $x < 0$ and $\theta(x) = 1$ for $x \geq 0$). Often, the bias $w_0$ is added as additional weight to $w$ with fixed input 1.

Learning in a perceptron means minimizing the error energy function shown above. This can be performed for example by gradient descent with respect to $w$ and $w_0$. This induces the well known delta-rule for the weight update

$$\Delta w = \eta(y(x) - t)x,$$

where $\eta$ denotes a chosen learning rate parameter, $y(x)$ the output of the neural network at sample $x$ and $t$ the observation of input $x$. It is easy to see that a perceptron separates the data linearly with the boundary hyperplane given by $\{x \in \mathbb{R}^n | w^T x + w_0 = 0\}$.

For the cell classifier, we use a single-unit perceptron with linear activation function in order to get a measure for the certainty of cell/non-cell classification. Application of delta-learning to the 5-dimensional data set from above gives excellent performance after already 4 epochs of batch learning. The final performance error (variance of perceptron estimation error of the training set) after 55 epochs was 0.0038 which confirms the good performance as well as the linearity of the classification problem. This was further shown, when we used a two-layered network with 5 hidden neurons in order to test for nonlinearities in the data set. After only 10 epochs, the error was already very small, and it could finally be diminished to $3 \cdot 10^{-12}$. Still the performance of the perceptron is more than enough for classification.

4 Confidence map

4.1 Generation

The cell classifier from above only has to be trained once.

Given such a cell classifier, section pictures can now be analyzed as follows:

A pixelwise scan of the image gives an image patch with center location at the scan point; to this image patch the cell classifier is then applied to give a probability of whether a cell is at the given location or not. This altogether (after image extension at the boundaries) yields a probability distribution over the whole image which is called confidence map. Each point of the confidence map is a value in $[0, 1]$ stating how probable it is that a cell is depicted at the specified location.

In practice a pixelwise scan is too expensive in terms of calculation time, so instead a grid value say 5 for $20 \times 20$-patches is introduced, and the picture is scanned only every 5-th pixel. This yields a rasterization of the original confidence map, which is still good enough to detect cells. Figure 4 shows the rasterized confidence map of a section part. The maxima of the confidence map correspond to the cell locations; small but non-zero values in the confidence map typically depict misclassifications that can be avoided by thresholding.

4.2 Evaluation

After the confidence map has been generated, it can be evaluated by simple maxima analysis. However as seen in figure 4, maxima not always correspond to cell positions, so thresholding in the confidence map is first applied. Values of 0.5 to 0.8 yield good results in experiments. Furthermore, the cell classifier could have responded to one cell when applied to image patches with large overlap. Therefore after a maximum has been detected, adjacent points in the confidence map are also set to zero within a given radius (15 to 18 were good values for $20 \times 20$ image patches). Iterative application of this algorithm then gave the final cell positions and hence the image segmentation.

5 Results

In practice we used perceptron learning after preprocessing with PCA and also ICA [5] [6] in order to help the learning algorithm with linearly separated data.

The patch size was chosen to be $20 \times 20$, a thresholding of 0.8 was applied in the confidence map and the
Figure 4. The plot shows the confidence map with grid value 5 of the image part shown above.

The cut-out radius for cell detection in the confidence map was 18 pixels.

In figure 1 an automatically segmented picture is shown. We see good performances of the counting algorithm.

So far we only compared cell-numbers of sections, counted by the algorithm and by an expert. We get calculation errors of about 5%. In further experiments, we also want to compare cell positions, detected by the algorithm and by an expert.

6 Conclusion

We have presented a framework for brain section image segmentation and analysis. The feature detector, here cell classifier, was first trained on a given sample set using a neural network. This detector was then applied by scanning over the image to get a confidence map. Maxima analysis yields the cell locations. Experiments showed good performance of the classifier, however larger tests will have to be performed.

In future work, various problems will have to be dealt with. First of all the scanning performance should be increased in order to be able to use smaller grid values, which could significantly increase the classification rate. This could be done for example by using some kind of hierarchical neural network like a cellular neural network, see [9]. In typical brain section images, some cells not directly lying in the focus plane occur in a blurred fashion. In order to count those without counting them twice in two section images with different focus planes, a three-dimensional cell classifier could be trained for fixed focus plane distances. A different approach for accounting for non-focused cells would be to simply allow ‘overcounting’, and then to reduce doubles in the segmented images according to location. This seems suitable given the fact that cells do not vary greatly in size. Finally, sections typically span more than one microscope image. In order to count cells of the whole section some way of not counting cells twice in both image parts has to be devised. This could be done for example by using techniques from image fusion. Furthermore, often not the whole image but only parts of the section are to be counted; so far this choice of the ‘region of interest’ is done manually. We hope to automate this in the future by finding separating features of these regions.

7 Acknowledgements

F.J.T. and E.W.L. gratefully acknowledges financial support by the DFG and the BMBF.

References


1 Graduate college ‘nonlinear dynamics’
2 Project ‘ModKog’