Robust Open Field Rodent Tracking using a Fully Convolutional Network and a Softargmax Distance Loss

Marcin Kopaczka[1], Tobias Jacob[1], Lisa Ernst[2], Mareike Schulz[2], Rene Tolba[2], Dorit Merhof[1]

[1]Institute of Imaging and Computer Vision, RWTH Aachen University
[2]Institute of Laboratory Animal Science, RWTH Aachen University
marcin.kopaczka@lfb.rwth-aachen.de

Abstract. Analysis of animal locomotion is a commonly used method for analyzing rodent behavior in laboratory animal science. In this context, the open field test is one of the main experiments for assessing treatment effects by analyzing changes in exploratory behavior of laboratory mice and rats. While a number of algorithms for automated analysis of open field experiments has been presented, most of these do not utilize deep learning methods. Therefore, we compare the performance of different deep learning approaches to perform animal localization in open field studies. As our key methodological contribution, we present a novel softargmax-based loss function that can be applied to fully convolutional networks such as the U-Net to allow direct landmark regression from fully convolutional architectures.

1 Introduction and Previous Work

In translational medicine and neurological and psychatric research, experiments on animals are an important step during pre-clinical trials. During these experiments, a number of clinical and behavioral parameters are assessed to determine treatment effects. One of the most commonly used animal experiments is the open field test, in which the animal is placed into an open experimental environment and its exploration behavior is assessed. The animal's position, orientation and movement speed are recorded with a top-mounted camera and evaluated.
Since the open field test is a widely used experimental setting, a number of methods and tools have been introduced to allow automated quantitative analysis. Notable examples include the MiceProfiler [1], The Noldus EthoVision software [2] and the Live Mouse Tracker [3]. While well-established and robust, the image analysis methods used in these tools are using established image processing and pattern analysis algorithms which do not reflect current advances in deep learning research. Therefore, we will assess how different current advances in fully convolutional networks can contribute to a robust and computationally effective localization of rodents in open field experiments which is the key requirement for all subsequent tasks.

2 Materials and Methods

Here, we will describe all methods used for both ground truth generation and localization using deep learning methods. All research was performed on a set of open field recordings of single C57BL/6 mice which represent the most commonly used mouse strain in animal experiments.

2.1 Established methods for ground truth bootstrapping

Most current algorithms use well-established methods for animal localization such as background subtraction and morphological operations. While these methods are generally not as robust as modern approaches, their performance is usually sufficient for the task at hand due to the controlled environment in which the rodents are analyzed. We use these methods to create reference results that can also serve as automatically generated ground truth for our supervised deep learning algorithms which require annotated data for training. For our localization task, we require the centroid of the rodent which can be computed by segmenting the animal mask and computing the segmentation's centroid. To compute the segmentation, we perform background subtraction of the experimental box and thresholding followed by morphologically removing remaining noise. Subsequently, the largest remaining cluster is assumed to represent the mouse for which we compute the geometrical centroid.

2.2 Direct position regression using a VGG16 architecture

A common approach for image-based position regression is using a convolutional architecture to predict landmark positions. This is achieved by using a classification architecture and replacing the output layer with neurons for the x and y positions of the keypoint and subsequently training the net with a distance-based loss. In our case, we use the widely used VGG16 architecture [4] in which the last layer is adapted to predict x and y positions of the animal and train the network with an L1 loss. The network output can be directly interpreted as centroid coordinate prediction, thereby allowing end-to-end training with the centroid coordinates serving as target and eliminating the need for additional post-processing of the output.

2.3 Segmentation using a U-Net architecture

Since our pipeline generates a segmentation mask for centroid computation, we can use the images and their masks to train a fully convolutional network to generate masks in which we subsequently compute the segmentation centroids. The most widely used fully convolutional architecture for semantic segmentation is the U-Net [5], which we modified to use a lower number of channels due to the nature of the problem and used residual blocks in the convolutional layers. The network was subsequently trained with a cross-entropy loss to obtain segmentation predictions for frames from the video recordings.

2.4 Direct landmark regression with a U-Net and a softargmax loss

Fully convolutional networks such as the U-Net can be trained to directly predict landmark positions. This is achieved by a recently proposed specialized loss function that implements a differentiable approximation to the argmax function. [6]:

$$\textit{soft-argmax}(\mathbf{x}, \beta) = \frac{\sum_{i,j} \exp(\beta \mathbf{x}_{i,j}) \cdot \begin{pmatrix} i & j \end{pmatrix}}{\sum_{i',j'} \exp(\beta \mathbf{x}_{i',j'})}, \tag{1}$$

in which β is a heat map intensity factor and i, j are the pixel coordinates. For large β, equation 1 converges towards the *argmax* function

$$\textit{argmax}(\mathbf{x}) = \lim_{\beta \to \infty} \textit{soft-argmax}(\mathbf{x}, \beta) = \begin{pmatrix} i_{max}, & j_{max} \end{pmatrix} \tag{2}$$

where (i_{max}, j_{max}) is the coordinate of the pixel with the largest intensity in the heat map. By multiplying with the vector (i, j), we can establish a correspondence between pixel positions and actual coordinates.

However, training with just this loss function prooved numerically instable. Therefore, we present an extension of the above-described loss function to improve localization performance and to stabilize the training by inducing a hint to the actual mouse position. Our full loss for regressing the ground-truth position (y_0, y_1) is defined as:

$$\begin{aligned}
\mathbf{p} &= \frac{\exp\left(\mathbf{x} - \mathbf{x}_{max}\right)}{\sum_{i,j} \exp(\mathbf{x}_{i,j} - \mathbf{x}_{max})} \\
L_{sharpness} &= -\log \mathbf{p}_{y_0, y_1} \\
\tilde{y}_0 &= \sum_{i,j} \mathbf{p}_{i,j} \cdot i \\
\tilde{y}_1 &= \sum_{i,j} \mathbf{p}_{i,j} \cdot j \\
L_{distance} &= |\tilde{y}_0 - y_0| + |\tilde{y}_1 - y_1| \\
L &= L_{sharpness} + L_{distance},
\end{aligned}$$

where $\mathbf{x}$ is the two dimensional net output. $\mathbf{p}_{i,j} \in [0, 1]$ is interpreted as probability of the mouse being at a certain position (i, j). The estimated position including subpixel interpolation is $(\tilde{y}_0, \tilde{y}_1)$. The subtraction of $\mathbf{x}_{max}$ guarantess the result of the exponentiation to be finite. $L_{distance}$ is based on the softargmax loss [6]. This function is distance-aware and therefore well suitable for localization training. It penalizes probabilities based on their distance to the ground truth positions. The $L_{sharpness}$ term is based on the cross entropy loss, except normalizing over pixels not the channels. Additionally, it improves training speed and robustness by directly enlarging the probability of the correct position. We compute the loss at two positions in the network, once at the bottleneck stage and once at the end of the upsampling stage as our preliminary experiments have shown that this approach increases training time and prediction precision.

The overall loss which is backpropagated through the network is the sum of both partial losses. The full architecture is shown in Fig. 1.

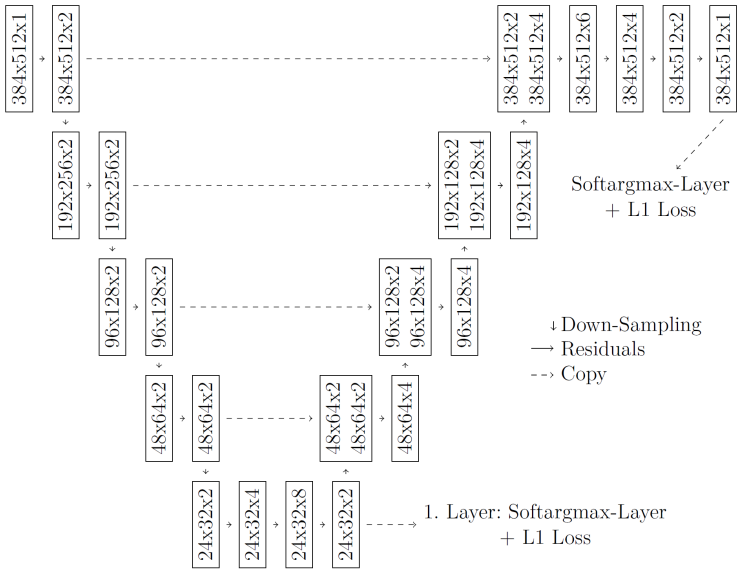

Fig. 1. Our proposed U-Net architecture with two instances of the distance and sharpness loss applied at the bottleneck and final stages.

3 Experiments and Results

To evaluate our loss function, we first generated a ground truth of segmentations and centroids from a set of 70 videos comprising 10000 individual frames departed in 10 subsets. Subsequently, all deep learning methods were validated using ten-fold cross validation.

Fig. 2 shows the results of the three implemented architectures. The basic VGG architecture delivers the least stable results, while both U-net-based approaches show a clearly higher precision. Results show that our proposed combined loss improves the detection accuracy beyond the default cross-entropy-based U-Net architecture while at the same time allowing direct training using landmark positions as ground truth, thereby eliminating the need for additional postoprocessing steps to obtain the centroids from the segmentation masks. The median error of our proposed architecture is 1 pixel on a 512 x 384 pixel input and a mouse-size of 35 pixels. In contrast, the cross-entropy-based U-Net achieves a median precision of 1.5 pixels and the VGG regression architecture achieves 5 pixels.

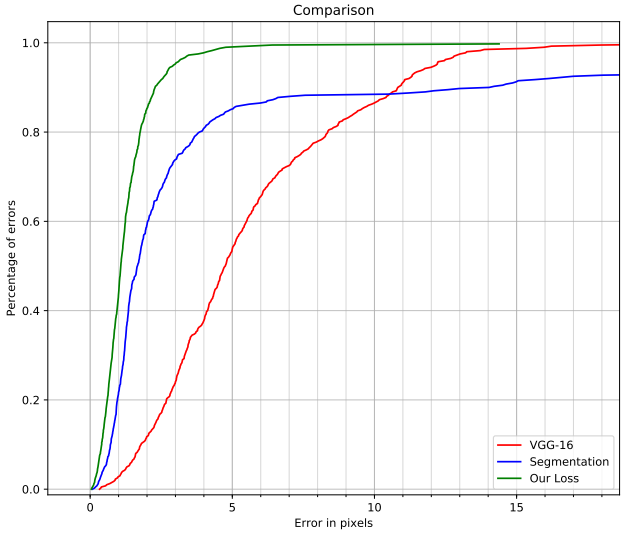

Fig. 2. Quantitative results of the three implemented architectures. The error is measured as pixel distance to the automatically generated ground reference. Lower values are better.

An analysis of the results of the two loss functions shows that already the loss that is applied to the bottleneck stage allows precise detection which is subsequently refined by the upsampling stages. Sample results are shown in Fig. 3.

4 Discussion and Outlook

We evaluated a set of different deep learning methods for automated rodent localization in open field scenarios. Next to established architectures, we introduced a novel loss function that allows direct coordinate localization using fully convolutional architectures. Results show that our proposed approach outperforms other common approaches in terms of localization accuracy. Future work will include enhanced behavior analysis of rodent locomotion. Furthermore, we will investigate extending our architecture towards other tasks in medical imaging, for example multi-position and multi-class localization.

5 Acknowledgements

This research was fully funded by the German Research Foundation (DFG), project ID: ME 3737/18-1 and 651874.

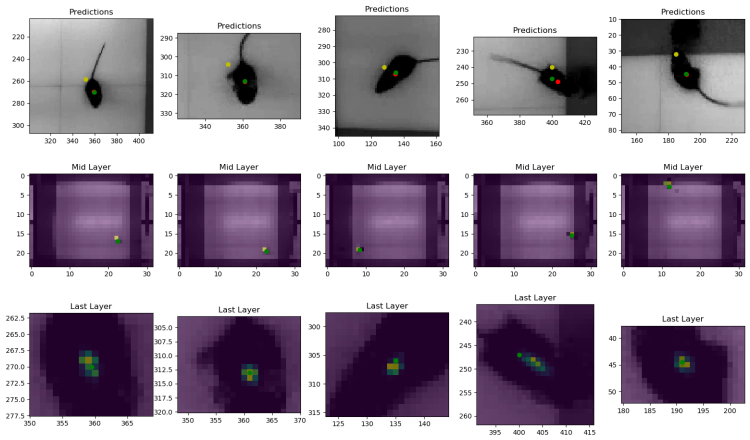

Fig. 3. Softargmax loss results. From top to bottom: final positions $(\tilde{y}_0, \tilde{y}_1)$ overlays on zoomed-in crops from the output images, map overlay of the bottleneck stage, map overlay on zoomed-in full resolution images. Yellow: positions predicted at the botleneck stage on the downsampled input size. Red: fully upsampled, final result. Green: ground truth (y_0, y_1). Best viewed electronically.

References

1. De Chaumont F, Coura RDS, Serreau P, Cressant A, Chabout J, Granon S, et al. Computerized video analysis of social interactions in mice. Nature methods. 2012;9(4):410.
2. Noldus LP, Spink AJ, Tegelenbosch RA. EthoVision: a versatile video tracking system for automation of behavioral experiments. Behavior Research Methods, Instruments, & Computers. 2001;33(3):398–414.
3. De Chaumont F, Ey E, Torquet N, Lagache T, Dallongeville S, Imbert A, et al. Live Mouse Tracker: real-time behavioral analysis of groups of mice. BioRxiv. 2018; p. 345132.
4. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014;.
5. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. p. 234–241.
6. Honari S, Molchanov P, Tyree S, Vincent P, Pal C, Kautz J. Improving landmark localization with semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 1546–1555.