



HAL
open science

Deep learning-based system for real-time behavior recognition and closed-loop control of behavioral mazes using depth sensing

Ana Gerós, Ricardo Cruz, Fabrice de Chaumont, Jaime Cardoso, Paulo Aguiar

► To cite this version:

Ana Gerós, Ricardo Cruz, Fabrice de Chaumont, Jaime Cardoso, Paulo Aguiar. Deep learning-based system for real-time behavior recognition and closed-loop control of behavioral mazes using depth sensing. 2022. pasteur-04163465

HAL Id: pasteur-04163465

<https://pasteur.hal.science/pasteur-04163465>

Preprint submitted on 20 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

1 TITLE

2 **Deep learning-based system for real-time behavior recognition and closed-loop control of**
3 **behavioral mazes using depth sensing**

4

5 Ana Gerós ^{1,2}, Ricardo Cruz ^{2,3}, Fabrice de Chaumont ⁴, Jaime S. Cardoso ^{2,3}, Paulo Aguiar ^{1,5*}

6 ¹ i3S – Instituto de Investigação e Inovação em Saúde, Universidade do Porto, Porto, Portugal;

7 Neuroengineering and Computational Neuroscience Group

8 ² FEUP – Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

9 ³ INESC TEC, Porto, Portugal

10 ⁴ Human Genetics and Cognitive Functions, Institut Pasteur, UMR 3571 CNRS, Université de Paris, France

11 ⁵ FMUP – Faculdade de Medicina da Universidade do Porto, Porto, Portugal

12

13 * For correspondence: [pauloaguiar@i3s.up.pt]

14

15 ABSTRACT

16 Robust quantification of animal behavior is fundamental in experimental neuroscience research.

17 Systems providing automated behavioral assessment are an important alternative to manual

18 measurements avoiding problems such as human bias, low reproducibility and high cost.

19 Integrating these tools with closed-loop control systems creates conditions to correlate

20 environment and behavioral expressions effectively, and ultimately explain the neural foundations

21 of behavior.

22 We present an integrated solution for automated behavioral analysis of rodents using deep

23 learning networks on video streams acquired from a depth-sensing camera. The use of depth

24 sensors has notable advantages: tracking/classification performance is improved and independent

25 of animals' coat color, and videos can be recorded in dark conditions without affecting animals'
26 natural behavior. Convolutional and recurrent layers were combined in deep network
27 architectures, and both spatial and temporal representations were successfully learned for a 4-
28 classes behavior classification task (standstill, walking, rearing and grooming). Integration with
29 Arduino microcontrollers creates an easy-to-use control platform providing low-latency feedback
30 signals based on the deep learning automatic classification of animal behavior. The complete
31 system, combining depth-sensor camera, computer, and Arduino microcontroller, allows simple
32 mapping of input-output control signals using the animal's current behavior and position. For
33 example, a feeder can be controlled not by pressing a lever but by the animal behavior itself. An
34 integrated graphical user interface completes a user-friendly and cost-effective solution for animal
35 tracking and behavior classification. This open-software/open-hardware platform can boost the
36 development of customized protocols for automated behavioral research, and support ever more
37 sophisticated, reliable and reproducible behavioral neuroscience experiments.

38

39 INTRODUCTION

40 Behavior is shaped by interactions between the organisms and the environment, being the most
41 important output response of the nervous system to external (and internal) stimuli. Understanding
42 this relationship between behavior and neural activity is the central goal of systems neuroscience,
43 which relies on analyzing animal behavior for theorizing cognitive mechanisms and ultimately
44 explaining the underlying neural circuits¹⁻³. Besides basic neuroscience research, the study of
45 animal behavior plays a key role in the translational analysis of disease models, preclinical
46 assessment of therapies' efficacy, and also in food production industries³.

47 The research on animal behavior has benefited from the recent technological advances in machine
48 vision and machine learning fields, allowing for the collection and automatic quantification of vast

49 amounts of data. Besides reducing human bias and subjectivity, and consequently allowing for the
50 standardization of measurements across laboratories, behavioral patterns that were once
51 unnoticed to a human observer may now be explored at different scales and resolutions⁴⁻⁶. The
52 first approaches to successfully combine computer vision and machine learning techniques
53 typically relied on hand-crafted features extracted from images or video sequences that can be
54 then used for automated behavior classification using supervised⁷⁻¹⁰ or unsupervised¹¹⁻¹³ learning
55 methods. However, such approaches are highly dependent on domain expertise for feature
56 engineering, often losing their generalization capability in the presence of a new
57 environment/scenario. Recent developments in the computational neuroscience field have
58 explored deep learning techniques to meet this challenge. Most state-of-the-art systems present
59 powerful deep learning-based solutions for pure body-part detection and tracking for pose
60 estimation¹⁴⁻²⁰, but modest progress has been made for direct recognition of behavioral events²¹⁻
61²³. When compared to action detection in humans, which already achieved outstanding
62 performance in challenging benchmarks, animals' behavior is more complex to characterize. First,
63 some animal behaviors are very similar to each other (more easily confused than those of
64 humans), in which temporal information is necessary for a flawless detection (sometimes a single
65 frame is not enough to label the behavior correctly). Recent approaches take advantage of deep
66 architectures that integrate temporal information along with spatial information to this end²¹⁻²³.
67 Also, different behaviors have different durations and temporal scales: some of them take place in
68 long time scales, such as *grooming*, and others in short time scales, such as *rearing* or *walking*. To
69 the best of authors' knowledge, temporal multi-scale integration has not been explored in the
70 context of animal behavior analysis. Another concern when planning behavioral experiments is to
71 ensure that the environment where the animal moves is adequate to allow capturing natural
72 behavior and yet probing for multiple parameters for its study. In particular, an important limiting

73 factor for recording natural rodent behavior is the environment lighting conditions (which may
74 affect animals' biological cycle). Usually, the most natural conditions are left behind at the
75 expense of recording conditions (higher image resolution or contrast). One possible strategy is to
76 use cameras with infrared technology (such as deep sensing cameras). A few studies have recently
77 begun combining deep learning methods with data from such technologies for animal behavior
78 analysis ²⁴. Finally, to effectively correlate behavioral functions with specific neural circuits,
79 automatic behavioral analysis tools should ideally be integrated into real-time closed-loop control
80 systems, that provide instantaneous feedback based on the current behavioral expression. There
81 are already published tools that provide feedback control in real-time based on animal posture
82 patterns ^{9,17,24-27}. However, they do not satisfy all these requirements simultaneously for a
83 complete and versatile behavioral analysis system.

84 Here, we introduce a novel computational solution for automated, markerless, real-time three-
85 dimensional (3D) tracking and behavior classification of 4 classes (*standstill*, *walking*, *rearing* and
86 *grooming*) in experiments with a single freely-behaving rodent. Combining the power of low-cost
87 depth sensors and deep learning techniques, the proposed framework is integrated into a control
88 platform that streams real-time mapping of input-output signals using the animal's current
89 behavior and position. First, we analyze the performance of advanced action recognition deep
90 learning networks on the rodent behavior dataset. Acknowledging the importance of integrating
91 temporal information in behavioral feature learning, we hypothesized whether abstract
92 spatiotemporal features obtained from simple deep networks are suitable for recognizing multiple
93 behaviors. In particular, the behavior of networks for increasing temporal extents and with
94 multiple timescales' branches (partially inspired in Feichtenhofer, et al. ²⁸) was compared
95 regarding their performance in detecting behavioral events. We found that temporal information
96 from the past, using a short-time scale, is most relevant for the learning process. Second, we

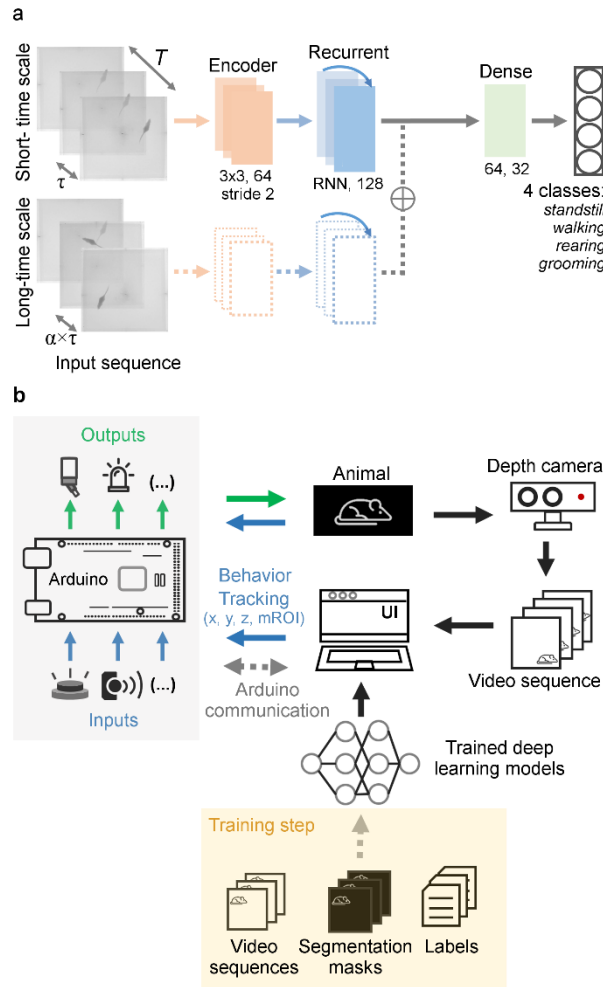
97 analyze how robust the proposed networks were at different input representations (input frame
98 encodings, sampling rates, and resolutions), where raw depth frames at higher sampling rates and
99 resolutions helped improve classification performance. Also, ~21 minutes (min) of annotated video
100 showed to be already sufficient to attain a good generalization using proposed deep networks for
101 behavior classification. Lastly, we adapt the deep learning framework to recognize animal tracking
102 and behavior in real-time, and we integrate it into a platform capable of closed-loop control of
103 behavioral experiments, either for behavioral mazes or real-time drug delivery systems. Besides
104 being non-invasive and with low latency, it provides a versatile interface to trigger different
105 hardware actuators from either hardware sensors or behavior/tracking-dependent signals.

106

107 RESULTS

108 The proposed system for online rodent behavioral recognition consists of two components: a deep
109 learning network (**Fig. 1a**) and a real-time control module (**Fig. 1b**). The network consists of an
110 encoder and a classifier, which is trained end-to-end. The encoder consists of two-dimensional
111 (2D) convolutional layers, to extract local spatial features in each frame of the video sequence. The
112 classifier is composed of a recurrent layer to learn temporal features between adjacent frames in
113 the video sequence, and a fully-connected layers to output the behavioral classes' probabilities
114 (**Fig. 1a**). Networks with different architectures and input representations were studied. Whereas
115 the deep learning network is responsible for spatiotemporal feature extraction and behavior
116 detection, the real-time classification is used to control sensors/actuators in any maze. All these
117 tasks can be controlled through an easy-to-use graphical user interface (GUI) for beginning-to-end
118 management of all experiments.

119



120

121 **Fig 1. Integrated framework for the control of behavioral mazes using depth information and deep learning-based**

122 **techniques. a.** Deep learning architecture, with the two variants of the encoder, single-branch (solid line) and dual-

123 branch (solid and dashed lines), for the automatic classification of 4 behavioral classes. Both variants receive one input

124 sequence with a time-window of size T ms, with frames equally spaced over time by a temporal stride of τ . The dual-

125 branch variant receives additionally one sequences with a different temporal stride, long-time scale pathway, that

126 operates on a bigger time-window ($\alpha \times T'$) with a temporal stride of $\alpha \times \tau$ ($\alpha > 1$, where α is the frame rate ratio between

127 short- and long-time scale pathways). **b.** Workflow of the closed-loop feedback system, for controlling behavioral

128 experiments. Depth video sequences are acquired by a depth camera, and used as inputs to deep learning networks for

129 real-time automatic classification of behavior and detection of animal's position (x, y, and z coordinates of centroid, and

130 any defined regions of interest inside the maze (mROI)). Such signals, together with input signals coming from any

131 sensor hardware (blue), are sent to the Arduino microcontroller for feedback control of the actuators present in the

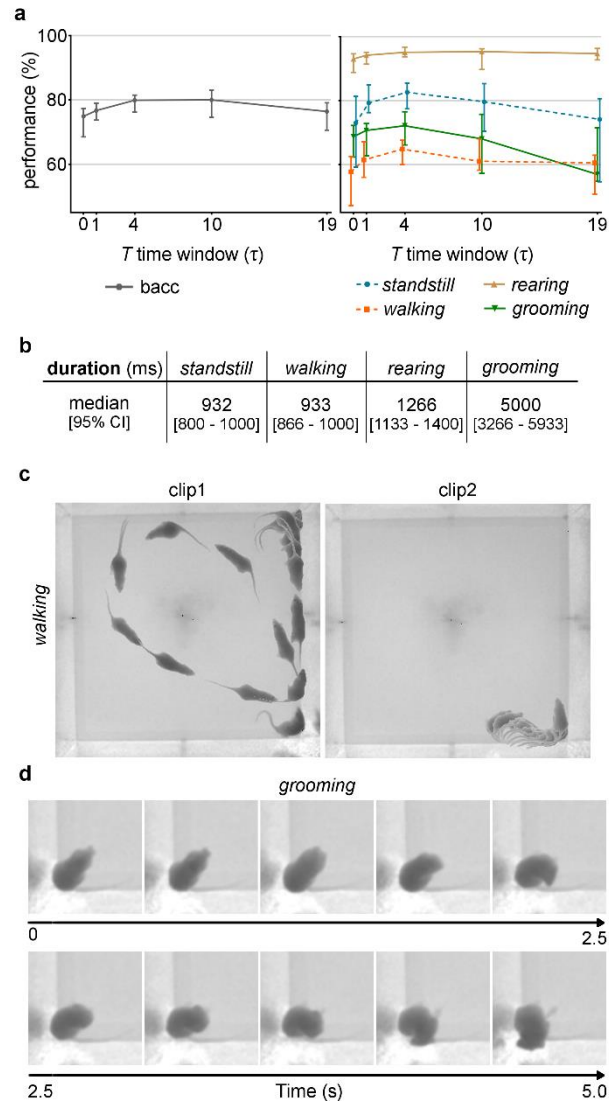
132 maze (green). For real-time behavior classification and detection of animal's position, the deep learning models must
133 first be trained using a training set with annotated depth video sequences (segmentation masks and behavioral labels).
134

135 **Past information improves behavioral classification performance**

136 To investigate the behavior of networks for increasing temporal extents, the time-window T of the
137 sliding input sequences was systematically increased, with a fixed temporal stride $\tau=133$ ms (**Fig.**
138 **2a** and **Supplementary Figure 1**). Improvements over T were observed, where models with a time-
139 window of 10τ (approximately 1500 ms, 11 frames in the sequence) achieved the top overall
140 results on the validation set, with a balanced accuracy of 80.0% [74.6, 83.0]%. No statistical
141 differences were found when using as input a time-window of 4τ . The results seem to indicate
142 that the gain of increased time-window is clearer for networks with a smaller time-windows, with
143 a converging trend towards time-windows above 1000 ms. This is aligned with the timescale for
144 the analyzed animal behavior classes (where the timescale for variation is in the order of 1 second)
145 (**Fig. 2b**). For time-windows smaller than 300 ms, the performance significantly dropped. When no
146 temporal information was taken into consideration, using a model with only one input frame, the
147 lowest overall accuracy was achieved, as well as category F1-score, showing that not only spatial
148 information within a particular frame may be important but also its motion content across
149 different frames. In fact, when performing manual annotations, ethologists often need to double-
150 check previous frames to annotate the current one, which also seems to happen in these
151 networks.

152 Out of all 4 classes, no behavioral event has a monotonic decrease with the increasing temporal
153 extent, and overall their recognition seems to benefit from time-windows smaller than 1000 ms
154 (category F1-score systematically increasing over T , until approximately 1000 ms). This effect is
155 particularly clear during *standstill*, *walking* and *grooming* events, where F1-score performance

156 seems to slightly decrease for time-windows greater than 1000 ms. In fact, *standstill* and *walking*
157 are events that usually last for a shorter period of time, compared to other behavioral events,
158 containing approximately 932 [800 – 1000] ms and 933 [866 - 1000] ms as median duration (**Fig.**
159 **2b**). For this reason, they do not seem to benefit from long time-windows for accurate recognition.
160 Furthermore, *walking* is the class with the lowest overall performance and one possible
161 explanation could be the fact that *walking* is the class containing greater intra-class movement
162 variability (either in terms of complexity of geometric shapes, sequences' durations and
163 movement speeds) (**Fig. 2c**). The behavioral event that appears to be the most sensitive one to
164 increasing the temporal extents is *grooming*. Using manual annotations given by the ethologists,
165 this action is typically composed of several stationary periods interspersed with shorter periods of
166 movement, in which the animal changes its position momentarily without leaving the *grooming*
167 event. Long-term networks, with larger time-windows, can, thus, easily confuse *grooming* with
168 *standstill* events (not shown), due to this heterogeneity within one single *grooming* sequence (one
169 example is shown in **Fig. 2c**, where a sequence of *grooming* frames was sampled at every 500 ms).
170 On the other hand, *rearing* is the class with the highest performance for the different time-
171 windows studied, not seeming to benefit from the increase in temporal extents. In fact, this is the
172 less ambiguous behavior in the current classification task, because of its easy-to-distinguish
173 geometric shape and lower depth values, and usually it is enough to analyze closer frames to
174 confirm it.
175



176

177 **Fig. 2. How much temporal information does the network need for rodents' behavioral learning?** **a.** Results using
 178 single-branch architecture of varying temporal extents. Left: Overall balanced accuracy (bacc) for increasing temporal

179 extents. Right: F1-score per class. Time window T in units of τ ($\tau = 133$ ms). Data represented as median \pm 95%

180 confidence interval (N = 5 trials). **b.** Behavioral events' duration, in milliseconds (ms). Data represented as median \pm 95%

181 confidence interval. **c.** Stroboscopic montage in which each animal position represents raw depth frames extracted at

182 every 266 ms for 2 different *walking* clips. **d.** Sample clips with frames extracted at every \sim 500 ms, for a single *grooming*

183 clip.

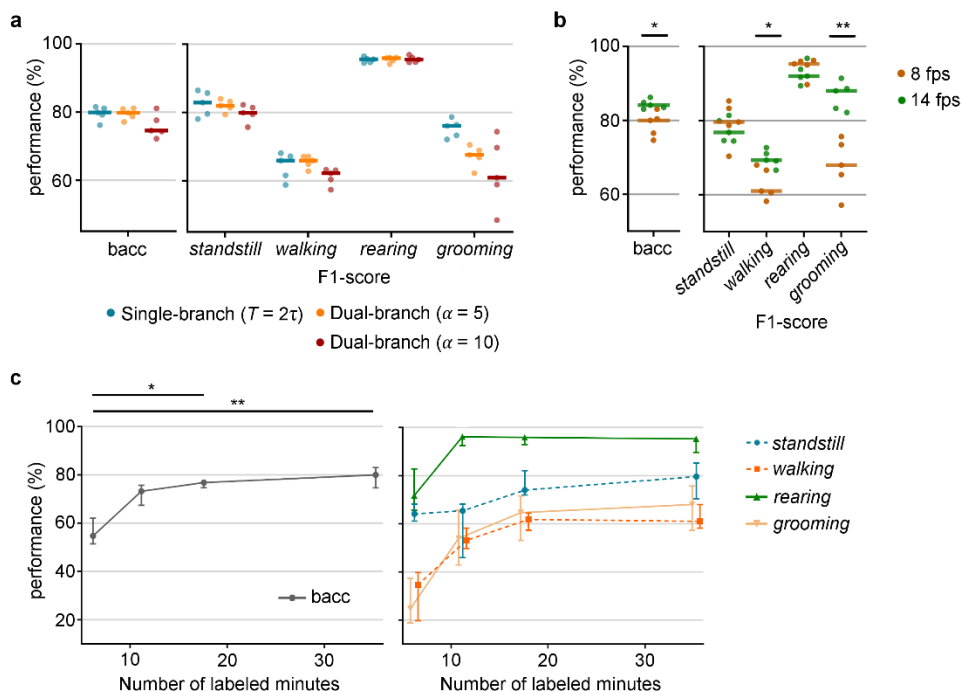
184

185 Short-time scales are the most relevant for the learning process

186 Additionally, two variants of network encoder, single- and dual-branch, were systematically
187 compared to study the impact of having temporal information of different scales. While in the
188 standard single-branch networks the input is a time-sliding sequence of size T ms, with a fixed
189 temporal stride τ ms between frames, this dual-branch network is fed with input sequences with
190 different temporal strides in each pathway, as a way to understand if having multiple time scales
191 helps in the learning process (**Fig. 1a**). The idea is for the two pathways to exploit temporal
192 information of a different scale: the short-time scale provides information hidden in temporally
193 neighboring frames, giving clues about animal's movement at fast temporal changes, while the
194 long-time scale may help distinguish between different behaviors at slower temporal changes
195 (namely, transitions between behavioral states).

196 To allow direct comparison, a single-branch architecture, with a time-window of 2τ and a
197 temporal stride of 133 ms, and a dual-branch architecture, with different frame rate ratios α
198 between the short- and long-time scale pathways, were trained and validated. The single-branch
199 and dual-branch $\alpha = 5$ appear to have similar overall performances (**Fig. 3a**), even for per-class
200 recognition; however α equal to 10 (which means doubling the time-window for that pathway)
201 seems to decrease performance. These results are in line with the conclusions of the previous
202 section, where behavior learning does not seem to benefit from very distant temporal information
203 (irrelevant frames are being taken into consideration, degrading network's performance).

204



205

206 **Fig. 3. a.** Which time scales are most relevant for the learning process? Comparison between architecture with different
 207 temporal scales: single-branch and dual-branch ($\alpha = 5$ and $\alpha = 10$), regarding overall balanced accuracy (bacc), and F1-
 208 score per class. **b.** How should time be distributed to increase performance? Comparison between different temporal
 209 strides τ between adjacent frames ($\tau \in \{67, 133\}$ ms, corresponding to approximately 15 or 8 frames sampled per
 210 second, respectively). **c.** How much information does the network need to learn? Overall and per-class classification
 211 performance as function of number of labeled minutes. Data represented as median \pm 95% confidence interval (N = 5
 212 trials). * $p < 0.05$; ** $p < 0.01$. Statistical analysis only for overall balanced accuracy for the sake of readability.

213 Additional statistical analysis on [Supplementary Figure 2](#).

214

215 Different input sequence's representations improve networks' learning

216 To further understand whether the temporal extent of video input sequences or their sampling
 217 frame rate with which the network is fed has more impact on learning rodents' behavior, networks
 218 with different temporal strides τ , but a fixed time window $T = 10\tau$, were also compared (**Fig. 3b**).
 219 Significant improvements were observed when using higher frame rates (smaller temporal
 220 strides), with an increase of approximately 5% in the overall performance (with a frame rate equal

221 to 15 fps, the median balanced accuracy reached 84.1% [83.0 - 86.2]%. In particular, *walking* and
222 *grooming* events greatly benefit from increasing the input frame rate. This could indicate that a
223 higher temporal resolution is needed to detect movement oscillations inherent to these types of
224 heterogeneous behavioral events.

225 As part of the networks' systematic study, the effects of input resolution and input depth encoding
226 were also examined. The highest resolution (256x256) achieved the best results, with an overall
227 performance of 85.9% [82.8 – 86.6]%. All behavioral events seem to benefit from increased
228 resolution, in particular *grooming*, with an increase of approximately 44% over the lowest
229 resolution (**Supplementary Figure 3A**). When changing input depth encoding, networks trained
230 with raw depth frames outperformed any other depth encoding techniques, with surface normal
231 inputs reporting the worst performance, yielding an overall accuracy of 71.8% [60.9 - 75.8]%.
232 (**Supplementary Figure 3B,C**).

233

234 [High performances achieved with a reduced training dataset](#)

235 In order to determine the approximate amount of annotated training data required for good
236 network performance, the size of the training set was systematically varied (**Fig. 3c**). As expected,
237 overall performance increases for increasing number of training images. Even 10k labeled frames
238 (approximately 21 min of labeled data) were enough to achieve a good generalization, above 70%,
239 with performance degradation in *walking* and *grooming* events. In fact, the effect of changing
240 training size is most significant in these classes, where increasing 20 min of annotated data leads
241 to a gain of almost 45% in per-class performance. Peak performance was reached with 30k training
242 examples (corresponding to approximately 1hour of labeled data).

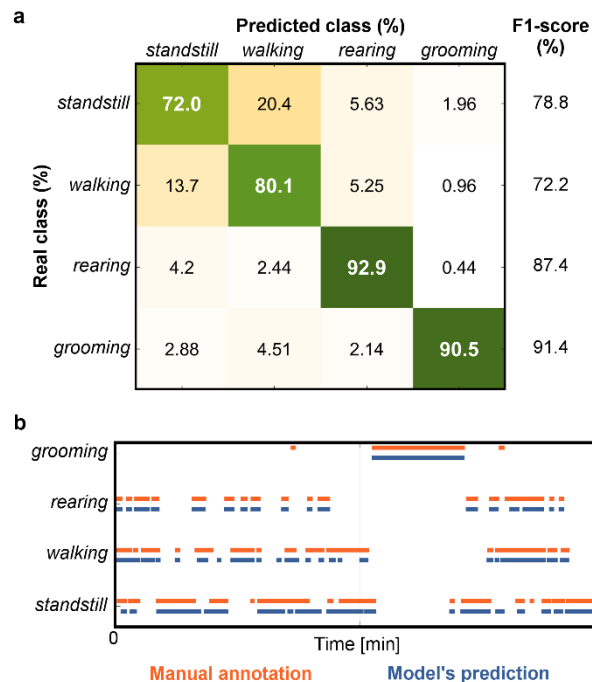
243

244 Behavior is accurately detected in unseen depth videos

245 The behavior of the network against a completely unseen testing set is the ultimate study to
 246 quantify recognition performance and generalization capability of the model (Fig. 4 a,b). After
 247 being trained with the best set of parameters, the model achieved an overall accuracy of 82.2 %
 248 [78.5 – 83.9]%. Together with the ethograms automatically generated (Fig. 4b), these results
 249 indicate that the proposed automated classification method captured the overall patterns of
 250 behavior in the new videos.

251 Regarding per-class performance, *rearing* is the behavioral event with the highest performance,
 252 attaining 87.2% [86.0 – 91.1]% F1-score, in accordance with previous results. Also, *walking* periods
 253 belong to the most misclassified behaviors, which are occasionally classified as *standstill* events
 254 (example in Fig. 4a), given frames' heterogeneity on shape and speed.

255



256

257 **Fig. 4. How does the best network behave for an unseen test set? a.** Example of normalized confusion matrix for a
 258 detailed analysis of automated behavior recognition errors, and corresponding F1-scores for each class. **b.** Example of

259 ethogram for a comparison between automated model's detection (orange) and manual annotation (blue), over 5 min
260 of testing video.

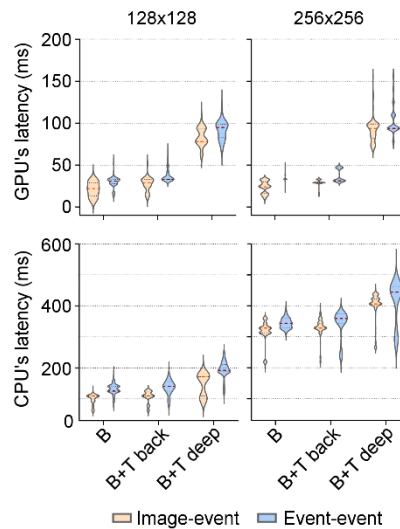
261

262 Closed-loop system achieves low-latency feedback based on animal

263 behavioral/tracking patterns

264 In order to create a system capable of controlling a behavioral task based on animal
265 behavior/position, it is necessary to close the loop between automatic detection of behavioral
266 events and experimental operant conditioning hardware. A control platform, combining a depth-
267 sensor camera, computer and Arduino microcontroller was constructed to allow mapping of input-
268 output control signals using the current deep learning detection of animal behavior and position.
269 Additional results on the performance of the segmentation task using deep networks can be found
270 in **Supplementary Figure 4**. To demonstrate the applicability of the closed-loop framework in
271 triggering signals based on animal behavior, an experiment was designed in which four actuators
272 (in this case, LEDs) were turned on when the rat performed one of the four behavioral events:
273 *standstill*, *walking*, *rearing* and *grooming*. The behaviors and tracking positions were automatically
274 detected by previously trained deep networks, that, together with input signals coming from
275 different sensors, are sent to the Arduino board to control the output devices. This setup achieved
276 delays from image acquisition to detecting the behavior+tracking position (image-event delay) as
277 fast as 28.9 ms [26.95 – 31.86] ms, for an input resolution of 128x128 (**Fig. 5a**). For larger images
278 (256x256), the delay increased about 8.9% (full results from additional configurations can be
279 found in **Fig. 5a**). The proposed system, with the advanced hardware configuration (GPU settings)
280 and for the smaller resolution, reached a performance time of 32.9 ms [32.8 – 34.9] ms from
281 predicting one behavioral event+tracking position to the next one (event-event delay), including
282 Arduino output generation, frame acquisition and processing, and behavior/tracking position

283 detection. Finally, sending the signal to the Arduino board and sending back the signal to the
284 computer took an additional 0.457 ms [0.457 – 0.460] ms, when compared to just turning on the
285 LED – event-LED delay (0.914 ms [0.913 – 0.914] ms). Thus, the Arduino response is not
286 constraining the runtime from event detection in one frame to the next frame, and it can be
287 almost entirely attributed to intrinsic camera frame rate, behavior/tracking detection and
288 additional processing.



289
290 **Fig. 5. How to close the loop for behavioral experiments?** Latencies, in milliseconds (ms), from image acquisition to
291 obtaining an event (image-event) and from the last event detected to the current event detected (event-event), using
292 CPU or GPU processing. Latencies were estimated for automated predictions of behavior only (B), behavior and tracking
293 using the background subtraction method (B + T back), and behavior and tracking using a deep model-based method (B
294 + T deep). The width of the violin plots represents the probability density of the data, with the median and 95%
295 confident interval represented as red and black dashed lines.

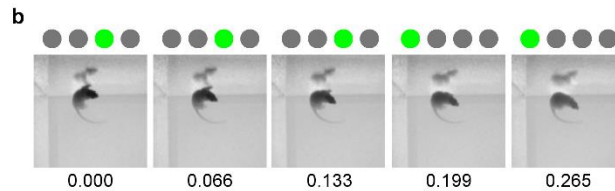
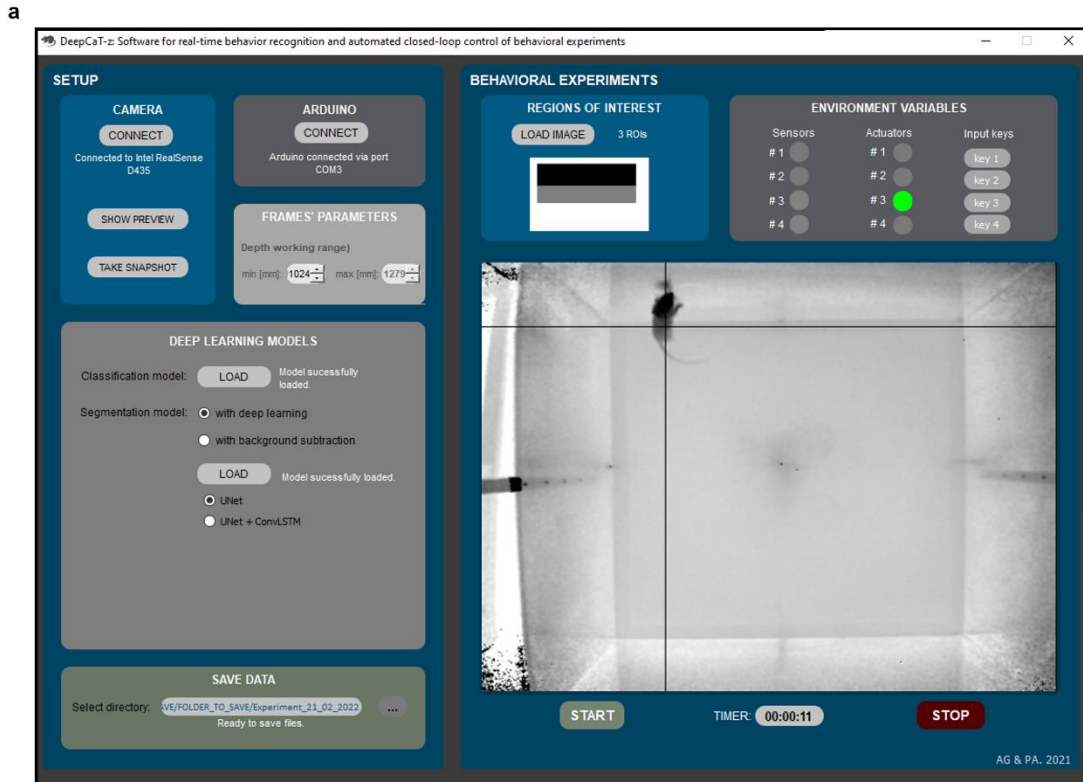
296

297 [User-interface allows end-to-end control of behavioral experiments](#)

298 Acknowledging the importance of embedding all algorithms in a user-friendly application suited
299 for research environments, we developed a full-featured, easy-to-use and freely available
300 software interface (**Fig. 6a**), requiring no programming by the end-user.

301 Behavior classification and/or tracking are performed using different available methods, chosen by
302 the user, and detected using uploaded trained models. The GUI provides online information
303 regarding hardware modules states, animal's behavior and position, allowing full control of the
304 entire system. In particular, the state of 4 sensors and 4 actuators are updated in real-time, in
305 which a LED-type icon is turned on upon the first image in which a behavioral pattern was
306 detected, and subsequently turned off upon the first image in which the pattern is no longer
307 detected (**Fig. 6b**). This allows for a fully closed-loop stimulus' framework. The GUI also includes an
308 option for users to upload an image containing ROIs for a more versatile and complete behavioral
309 analysis. All useful information recorded during the experiment (depth frames, tracking and
310 behavioral classes' information with sensors/actuators states for each timestamp) can be exported
311 to a user-defined directory for further analysis.

312 Overall, a cost-effective and easy-to-setup framework was created. The entire system consists of a
313 computer running the GUI, connected to a depth camera (e.g., Intel RealSense Depth Cameras, of
314 ~300 €) and an Arduino (e.g. Mega 250, of ~35 €). Sensors and actuators can be directly connected
315 to the Arduino board, and the quantity and type depend on each experiment's goal. The source
316 code of the software, together with the user-guide manual, list of hardware materials and video
317 examples, are publicly available for download at GitHub ([https://github.com/CaT-zTools/Deep-](https://github.com/CaT-zTools/Deep-CaT-z-Software)
318 [CaT-z-Software](https://github.com/CaT-zTools/Deep-CaT-z-Software)).



319

320 **Fig. 6. How to easily control behavioral experiments?** **a.** Graphical user interface for automating real-time closed-loop

321 behavioral experiments. **b.** Example of a *rearing* followed by a *walking* sequence, with corresponding LED status (as it

322 appears in the graphical user interface), from the test video sequence. Image timestamps in seconds are presented at

323 the bottom of each image.

324

325 DISCUSSION

326 We have presented a fully integrated framework that can provide real-time feedback based on
327 automated rodents' behavior classification and tracking position, using specialized deep neural
328 networks to extract information from frames acquired with depth-sensing technologies.

329 With the developed algorithms, we demonstrate that cutting-edge deep learning models can be
330 used to learn features from depth video sequences, without the need for feature-engineering
331 approaches. In fact, this is one of the main reasons why deep learning-based methods can be
332 more powerful than conventional behavior classification ones, avoiding user bias in the learning
333 process and allowing for more easily tunable and generalizable systems. This is particularly
334 important in basic research where environmental setups or animals' appearance/strains may be
335 changed depending on the objectives of each experiment and yet it is possible to successfully
336 apply the same methods ^{3,6}.

337 Furthermore, the capabilities of these deep learning networks were extended to learn feature
338 representations exclusively from depth information. Although several deep learning-based studies
339 have been published using depth frames for detecting human behavior, depth information is
340 usually incorporated using multi-branch architectures, combining color and depth inputs from
341 multiple streams for motion capture ²⁹⁻³¹. Here, we focused on depth images and how information
342 can be successfully retrieved for animal behavior extraction. Analyzing behavior with only depth
343 information has four important advantages. Since these frames are acquired by infrared sensors,
344 videos can be recorded in dark conditions (where color information is useless) without disrupting
345 animals' natural behavior (mainly in nocturnal animals, such as rodents). Also, with this
346 technology, color contrast between the animal and the background is no longer a problem for
347 detection/tracking purposes. Conventional methods usually use markers or methods dependent
348 on animals' color coating ³²⁻³⁶, which can be avoided using depth-sensing information. In addition,

349 3D information can be retrieved from a single camera, and so setting complicated stereo-vision
350 setups is no longer needed. Finally, to further facilitate the integration of computational methods
351 in the laboratory and industry fields, low-cost acquisition devices are required, combined with
352 good performance and, at the same time, quick data acquisition and low computational cost.
353 Therefore, the use of depth technology, such as *Kinect*-based cameras, showed to be an
354 alternative strategy to be applied in behavioral experiments. Since there are no state-of-the-art
355 studies exploring the use of depth information in the context of feature extraction for animal
356 behavior classification, we also perform a systematic study to understand the best ways to
357 represent network inputs and how we can improve models' performance. By using deep learning
358 networks that incorporate spatiotemporal features, it was possible to conclude that temporal
359 information is very relevant for learning animal behavioral patterns, especially in some classes
360 (*standstill* and *walking*, which contain a strong dynamic component). These results are in
361 agreement with the fact that temporal information of video data can provide additional clues
362 hidden in temporally neighboring frames for the recognition of actions/behaviors or segmentation
363 of frames^{29,37}. By using a fixed temporal stride between input frames of approximately 133 ms,
364 the performance of networks is significantly improved for input video sequences with a time-
365 window of approximately 1.5 seconds. As expected, some animal behaviors are of very short
366 duration, with rapid transitions, sometimes imperceptible by humans, and for this reason, deep
367 neural networks for animal behavior classification must be carefully designed to support finer
368 temporal analyses. In addition, results showed that neither long-time scales nor multi-scales
369 seemed to be advantageous for detecting animal behavior. One possible explanation is that long-
370 time scales include frames too far apart in time, containing irrelevant information to learn useful
371 feature representations for the current frame. Although with our system we didn't see advantages
372 in the multi-scale analysis, we hope that it can be further explored in the context of animal

373 behavior. For example, in a system with higher frame rates, it may be useful to also explore
374 shorter time scales.

375 Along with the fact that higher resolutions and higher sampling rates in raw frames (without
376 preprocessing or encoding) significantly improve the performance of proposed deep networks, the
377 results give an insight on how to build, train and fine-tune networks to better learn rodent
378 behavior using depth-sensing information. Finding that ~21 min of annotated videos are already
379 sufficient to achieve high generalization rates strengthens the contributions of the proposed
380 system since a core goal of automating the analysis of behavior is reducing the manual annotation
381 effort. In this sense, once the deep learning model is trained, the system is ready to assist in any
382 behavioral experiment without additional user-time, allowing for more reproducible results and
383 reducing variability imposed by inter-human annotations. Recent works have made some progress
384 toward the goal of supervised classification of rodents' behavior using deep learning techniques to
385 improve conventional feature-engineering-dependent methods. Marks, et al. ²² developed
386 *SIPEC:BehaveNet* for behavior recognition, which was tested in a dataset acquired with a
387 conventional camera and containing freely behaving mice whose behavior was labeled with only 3
388 classes ³⁸. Although claiming superior performance to Sturman, et al. ³⁸ proposal, *SIPEC:BehavNet*
389 achieved lower overall performances for *supported rearing* and *grooming* events (mean \pm
390 standard error of the mean: 0.84 ± 0.04 and 0.49 ± 0.21 , respectively), when compared to what we
391 were able to report here. *DeepEthogram* is another recent tool for frame-based classification of
392 animal behavior in RGB videos ³⁹. High overall performances (overall accuracy) were obtained for
393 datasets containing mice behavior with more than 4 classes. However, performance per-class (F1-
394 score) is substantially impaired for some behaviors, in particular, the rarest and most challenging
395 behaviors in the dataset (average F1-score above 70%). This shows evidence that attention must
396 be paid to metrics performance when dealing with highly unbalanced datasets. Overall, both

397 methods fall behind some strengths that our method shows, needing more than 70 min of labeled
398 data to achieve a comparable performance (overall accuracy above 70%) and not being suitable
399 for natural environmental conditions in the analysis of rodents' behavior.

400 In order to improve the potential of the proposed system and create an integrated tool that would
401 boost future development in understanding behavioral patterns and neuronal activity relationship,
402 deep learning-based detection of behavior was used to provide event-triggered feedback in real-
403 time. The loop between animals' maze, depth frames acquisition, and automatic streaming of
404 behavioral patterns was closed using input and output devices connected to an Arduino
405 microcontroller. From detecting one behavioral event to the next event in a consecutive frame,
406 the system was able to achieve real-time feedback control, with latencies of less than 33 ms with
407 GPU-based configuration. These results are below the frame rate of the camera used (which
408 typically is reduced to ~15 fps in low light conditions), and so, in theory, more powerful infrared
409 cameras could be tested. Research on developing real-time applications for neuroscience research
410 has been advancing in recent years. However, efforts have essentially been directed towards tools
411 to detect animal's posture, rather than classifying directly the behavior. Both Forys et al., 2020 and
412 Schweihoff et al, 2021 developed software and hardware to enable real-time estimation of mice
413 posture, and achieved latencies of 30ms using comparable computational configurations, from
414 frame acquisition to detecting a posture of interest (slower image-event delay than what we were
415 able to achieve)^{17,26}. Kane, et al.²⁵ reported higher computational performances for the same
416 task, with a 16ms delay from image-LED event (for equivalent image resolution and hardware
417 configurations). However, it is worth emphasizing that our 30-fps figure is achieved when both
418 behavior classification and tracking position are available, which gives the tool versatility for
419 different research applications. To the best of authors' knowledge, Nourizonoz, et al.²⁴ were the
420 first to try to detect animal postures as well as simple behaviors in naturalistic environments, using

421 multiple cameras with infrared-based technology. Real-time detections were achieved to enable
422 reinforcing a simple behavior (*rearing*) by operant conditioning. Although with high performance
423 in naturalistic environments and taking the first steps in moving forward to correlate posture with
424 neural circuits by optogenetics stimulation, the detection of a single behavior from posture was
425 achieved using a set of geometrical rules. This approach may not be sufficient to classify more
426 sophisticated behaviors, or computationally heavier when classifying multiple behaviors.

427 A key aspect of the design of the whole system is its versatility and how different modules can be
428 adapted to different research goals. In particular, several tracking algorithms were made available,
429 depending on model's performance and computational power. This flexibility may be important
430 when real-time detection is not required but offline high-performance detection is needed. Also,
431 many sensors and actuators can be easily adapted to the Arduino microcontroller to finer control
432 of animal's maze, and the automation control code is prepared to be further extended. Even so,
433 recent advances in multiple animal behavior analysis and tracking^{9,16,40} could be included to
434 further enhance this versatility. System adaptation is, in theory, straightforward, however, the
435 triggers for feedback control need to be carefully designed when dealing with complex social
436 behavior. Furthermore, the list of behavioral events/classes can be further extended. Here, the
437 potential of deep neural networks can be explored, since they are able to extract relevant features
438 without the need for feature re-engineering, unlike conventional machine learning methods.

439 Taking all the contributions together, we believe that the flexibility and yet easy-to-use
440 characteristics of this real-time feedback framework may open the door to further studies and
441 broader applications, allowing more high-throughput and rigorous behavioral experiments while
442 less invasive for laboratory animals.

443

444 MATERIALS & METHODS

445 Dataset

446 An open-access RGB-D behavioral dataset, available at <https://doi.org/10.5281/zenodo.3636135>¹⁰,
447 was used for all experiments. Details on the experimental procedures, video acquisition and
448 manual annotation of rodent's behavior can be found in ¹⁰. In brief, the dataset is composed of 10
449 to 15 min RGB-D video sequences of individual Wistar rat behavior, recorded with a *Microsoft*
450 *Kinect v2* camera (512x424 depth pixel resolution). The maximum frame rate is 30 frames per
451 second (fps), but this value typically drops to 10 to 15 fps in low light conditions. A subset list of
452 classes was considered here with the four most commonly used state behavior states: *standstill*,
453 *walking*, *rearing* and *grooming*. A randomly selected subset of these fully annotated recordings
454 was considered for the experiments and denoted as *dataset-100k* (~2.20 h in 26 subvideos,
455 approximately 100,000 frames total, with a time difference between two consecutive frames of
456 approximately 67 milliseconds (ms)). Only the depth frames were kept for analysis.

457

458 Proposed deep learning model

459 Architecture

460 Two variants of the encoder were considered – the single-branch and the dual-branch. In both
461 architectures, frames are individually encoded by four 2D convolutional layers (64 filters, 3x3
462 kernel size, 2x2 stride, rectified linear unit (ReLU) activation). After the encoding part, a recurrent
463 layer (RNN, 128 hidden state features) takes as input the sequence of spatial features output by
464 the feature extractor and integrates it over time for both temporal and spatial dynamics learning.
465 Two fully-connected layers (64 and 32 filters) and a softmax output layer are used for the final
466 recognition of behavioral classes. In the case of the dual-branch, both pathways work on different
467 time-windows: the short-time scale pathway receives as input a pre-defined time-window T' with

468 the same temporal stride τ as the single-branch network; the long-time scale pathway operates
469 on a bigger time-window ($\alpha \times T'$) with a temporal stride of $\alpha \times \tau$, where $\alpha > 1$ is the frame
470 rate ratio between short- and long-time scale pathways. Two recurrent layers are used for each
471 branch, which are then concatenated before the fully-connected layers.
472 Since recognizing rodent's behavior is a challenging task, either due to the size of the animals or
473 the nature of the behaviors (faster movement, higher similarity and greatly dependent on
474 temporal information to be clearly distinguished), the feature extraction process needs to be
475 carefully designed to avoid confusion between behavioral events. For this reason, 2D convolutions
476 were chosen, instead of the currently used 3D convolutions for spatiotemporal learning, in order
477 to process spatial and temporal content separately and thus avoid mixing information of different
478 scales. The reduced number of convolutional layers and the number of filters at each layer allow
479 the entire network to be computationally lightweight and capable of being used for real-time
480 inference afterwards.

481

482 Training

483 The models were trained from scratch using the Adam optimizer, with a batch size of 16 video
484 sequences with a time-window of T ms, and a learning rate of 1×10^{-4} , for 100 epochs. A
485 dropout layer was used before the recurrent layer, with a dropout ratio of 0.5.
486 Initially, the dataset was split into training (70%), validation (10%) and testing (20%) sets that are
487 maintained throughout the experiments. The validation set was used to compare the performance
488 of different models when performing ablation studies. To address the problem of having a highly
489 imbalanced dataset (*standstill* 40.3%, *walking* 28.7%, *rearing* 11.7%, and *grooming* 19.3%), the
490 video sequences of each class were oversampled until their frequencies were uniform.

491

492 Experiments

493 For a systematic study of networks' performance, the effect of increased temporal information
494 was evaluated, by changing different parameters in each experiment. First, the impact of changing
495 the time-window T of the input sequence was tested, with $T \in \{0\tau, 1\tau, 4\tau, 10\tau, 19\tau\}$ ms,
496 corresponding to a network input with 1 (single-frame), 2, 5, 11 and 20 frames in total,
497 respectively, sampled with a fixed temporal stride τ of 133 ms. Also, the temporal stride τ
498 between adjacent frames ($\tau \in \{67, 133\}$ ms), was also evaluated, which corresponds to
499 approximately 15 or 8 frames sampled per second, with a fixed time-window. Finally, the frame
500 rate ratio α between short- and long-time scale pathways for the multi-branch architecture ($\alpha \in$
501 $\{5, 10\}$) was varied. These temporal parameters were chosen in order to make the network
502 responsive to the different behavior timescales present in the original dataset. In this sense, and
503 taking into consideration the camera's frame rate, the capability of the network of capturing both
504 fast behavioral events (in the order of a few hundred milliseconds) and slower events (in the order
505 of a few seconds) was explored. Also, different spatial resolutions of $\{64, 128, 256\}$ pixels and
506 input encoding modalities were tested. Besides raw 8-bit depth frames, depth jet-encoding⁴¹ was
507 applied to depth frames, in which the depth information is distributed according to the jet
508 colormap, transforming the one-channel depth map to a three-channel color image. Also, surface
509 normals were used to encode the depth frames into a three-channel image representing form and
510 surface structure (implementation details in⁴²). Unless otherwise noted, the full *dataset-100k* was
511 considered for analysis, and the default parameters for the systematic study were: $T = 10\tau$, $\tau =$
512 133 ms, spatial resolution of 128 pixels in raw depth frames. The influence of training set size on
513 network generalization was also benchmarked. Different training sizes were selected and each
514 subsampled training set was used to train the network, and compared with the same validation
515 set (using the default parameters' set as well).

516

517 Data augmentation

518 To improve the robustness and generalization of the models, data augmentation was performed
519 with random perturbations of the training set during training, that included: full-rotation around
520 the center (90/180/270°); horizontal flipping; resized cropping and brightness variation (by
521 sampling an additive value from a uniform distribution, [-0.15, 0.15]). As the input of all models is
522 a frame sequence of approximately T/τ frames, the same augmentation operations were
523 performed on each frame in this set.

524

525 Model evaluation and metrics

526 The validation set was used for models' comparison and evaluation, and all analyses reported
527 share the same validation set, for a total of 5 runs for each experiment. The hold-out testing set
528 was further applied to evaluate the performance of the best-chosen model to an unseen set. To
529 evaluate the overall performance of the different proposed methods, balanced accuracy (average
530 of recall obtained on each class) was calculated. Performance per class was assessed using
531 confusion matrices and corresponding F1-score.

532 The F1-score is the harmonic average of the precision and recall, calculated as follows:

$$533 \quad F1 \text{ score} = 2 \times \frac{\textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}},$$

$$534 \quad \textit{where precision} = \frac{\textit{true positive}}{(\textit{true positive} + \textit{false positive})} \textit{ and recall} = \frac{\textit{true positive}}{(\textit{true positive} + \textit{false negative})}.$$

535 These metrics are better suited to deal with imbalanced datasets.

536

537 Real-time control system

538 The entire control system consists of software and hardware modules configured to create an
539 automated closed-loop tool. It is made of five main components: the control computer, the
540 interface board, the control software, the video camera and the maze hardware modules (Fig. 1).
541 Frames acquired by a depth camera are fed into the trained deep learning models, which will

542 automatically detect both behavioral events and the animal's position in the maze. The network
543 outputs are sent to the interface board that, together with existing sensor outputs (e.g., buttons,
544 maze sensors), controls circuit actuators (e.g., maze feeders, light-emitting diodes (LED)s). The
545 computer is used to operate the entire circuit by a graphical user interface (GUI), either sending
546 messages to the interface board or acting directly on the maze hardware modules.

547

548 [Interface board](#)

549 An Arduino microcontroller (Mega 2560) was used as the interface board between the computer
550 and the hardware modules, and the communication is established using a communication (COM)
551 port. The microcontroller board has 16 MHz clock speed, and 54 digital input/output pins that can
552 be connected to different maze hardware components, such as animal feeders, LEDs, maze
553 sensors, and buttons. After being connected to the computer, the Arduino board communicates
554 via Arduino integrated development environment (IDE). The user writes the Arduino code for the
555 automated control in the IDE, uploads it to the microcontroller which executes the code to
556 interact with the input and output hardware modules. Notice that, once uploaded, the code can
557 run regardless of the connection between the Arduino and the computer.

558

559 [Control software](#)

560 The automated control software consists of the following components: the automation control
561 code, the trained deep learning models for detection, and the data acquisition and communication
562 protocol.

563

564 [Automation control code](#)

565 Arduino code is written within the Arduino IDE (in a language very similar to C++) and it was
566 carefully organized to segregate the code for specific logic state implementations (automated

567 control) from all other maintenance code (such as reading and writing data to the communication
568 port (COM). To do so, a specific user-defined function was created, which has access to all critical
569 variables for the control, such as sensors' and actuators' states, and animal's position and
570 behavior. Inside this function, the user can easily define the conditions of stimuli-response that
571 characterize each behavioral test experiment.

572

573 *Deep learning models*

574 In order to automatically classify the behavior and calculate the position of the animal using deep
575 learning methods, previously trained models are imported and directly used for predictions. For
576 the automatic classification of behavior, the single-branch model was trained according to the
577 protocol previously described (input sequence of raw depth frames, with a time-window of
578 approximately 1330 ms, acquired at a frame rate of 15 fps). For the estimation of animal's
579 position, two different methods were made available to the user: deep learning-based model for
580 semantic image segmentation, and conventional background subtraction model, both followed by
581 centroid calculation. The deep learning-based model combines two ingredients from deep
582 networks' knowledge in order to perform semantic segmentation taking into consideration
583 temporal information: U-Net model as backbone architecture, and (optional) convolutional Long
584 Short-Term Memory (ConvLSTM) layers, learn spatiotemporal features. The traditional U-Net
585 architecture was reduced to only one convolutional layer per block, fewer filters per layer (32) and
586 it was extended by placing two ConvLSTM layers, one between the encoder and the decoder, and
587 the other one before the last dense layer (different positions in the network, as well as different
588 architecture parameters, were tested to ensure maximum performance yet reduced inference
589 time and memory (**Supplementary Figure 4**)). The network was trained from scratch using 1220
590 train and 320 validation video sequences (previously annotated to obtain the segmentation
591 masks), with ADAM optimizer and dice binary cross-entropy loss function.

592 A conventional background subtraction method was integrated in parallel to provide a
593 computationally lighter alternative yet with lower performance (mainly in frames with dynamic
594 backgrounds). Using this method, the segmentation mask containing animal's pixels is produced
595 by subtracting the present frame with the background model (frame of the behavioral
596 experimental setup without the animal). From the segmentation mask, the position of the animal
597 is calculated as the centroid of the detected object/animal. For details on algorithm's design and
598 performance, please check Gerós, et al. ¹⁰.

599 For a more complete information about animal's movements inside the maze, the system allows
600 the user to define spatial regions of interest inside the maze (mROI), by uploading an image file
601 with the same resolution as the acquired frames, with the different mROIs painted uniformly with
602 different colors. Those regions are automatically detected after getting animal's tracking, and they
603 will be used as input for the Arduino board to control the hardware mazes, if needed.

604

605 *Data acquisition and communication*

606 To establish the communication between the COM port and the Arduino board, a communication
607 protocol was defined. The computer communicates with the interface board by sending the
608 behavioral classification, tracking and mROI outputs (as well as a flag for any keypress), in the form
609 of a characters' list separated by commas. Each character encodes information for the behavioral
610 state (*S* for *standstill*; *W* for *walking*; *R* for *rearing*, and *G* for *grooming*), tracking (*x*, *y* and *z*
611 coordinates of the centroid), mROI and a key-pressed flag (both encoded as integers). On the
612 other hand, the Arduino board sends information regarding the status of each of the sensors and
613 actuators (binary coded, on/off) back to the computer.

614

615 Video camera

616 The acquisition protocol was developed using a new generation of low-cost depth cameras, the
617 Intel® RealSense Depth Cameras (in particular, D435 model), acquired with 512x424 depth pixel
618 resolution and at a maximum of 30 fps.

619

620 Computational performance: inference and latency times

621 To test time-performance of the system, a video of a freely-walking rat was used to simulate a
622 camera feed from an animal in real-time, and single frames from the video were loaded at the
623 maximum rate of 30Hz. The bidirectional communication with the Arduino board was achieved
624 from either four input sensors and signals from the computer, and four output actuators (in this
625 case, LEDs). Three latency periods were measured: (a) the delay from image acquisition to
626 detecting the behavioral state/tracking position (image-event delay); (b) the delay from detecting
627 one behavioral event/tracking position to the next event/tracking position (event-event delay,
628 including *Arduino* response, mROI detection, GUI updates and saving images to external folder); (c)
629 the delay between sending a behavioral state to the Arduino and turn on the corresponding LED
630 (event-LED delay, with and without output feedback of Arduino). The first two latency times were
631 determined using software timestamps and the last one was measured using the oscilloscope.

632

633 Computing hardware

634 All experiments, including inference speed and feedback control tests, were conducted on an Intel
635 Core i9-7940X (128 GB RAM), and a NVIDIA GeForce RTX 2080 graphics processing unit (GPU) (8
636 GB RAM), running *Windows 10*, with *Python 3.9* using *PyTorch* (1.8.1) and *TensorFlow-GPU* (2.5.0)
637 frameworks. All algorithms were integrated into a user-friendly GUI, designed in the *Qt Creator*
638 (*The Qt Company*, Finland) environment and implemented in *Python* language.

639

640 [Statistical methods](#)

641 Statistical analysis was performed using GraphPad Prism version 7.00 (GraphPad Software Inc., CA,
642 USA). The method of D'Agostino & Pearson was used as a normality test, and parametric or non-
643 parametric tests were chosen as appropriate. Statistical significance was considered for $p < 0.05$.
644 Parametric data are expressed as mean \pm standard deviation (SD), and non-parametric data are
645 expressed as median and 95% confidence intervals.

646

647 [Data availability](#)

648 The open-access RGB-D behavioral dataset used for all experiments is available at
649 <https://doi.org/10.5281/zenodo.3636135>.

650

651 [Code availability](#)

652 The source code of the software, together with the user-guide manual and list of hardware
653 materials, are publicly available for download at GitHub ([https://github.com/CaT-zTools/Deep-
654 CaT-z-Software](https://github.com/CaT-zTools/Deep-CaT-z-Software)).

655

656 References

- 657 1 Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A. & Poeppel, D.
658 Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron* **93**, 480-490,
659 doi:10.1016/j.neuron.2016.12.041 (2017).
- 660 2 Berman, G. J. Measuring behavior across scales. *BMC Biol* **16**, 23, doi:10.1186/s12915-018-
661 0494-7 (2018).
- 662 3 Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* **84**, 18-
663 31, doi:10.1016/j.neuron.2014.09.005 (2014).
- 664 4 Robie, A. A., Seagraves, K. M., Egnor, S. E. & Branson, K. Machine vision methods for
665 analyzing social interactions. *J Exp Biol* **220**, 25-34, doi:10.1242/jeb.142281 (2017).
- 666 5 Macpherson, T. *et al.* Natural and Artificial Intelligence: A brief introduction to the interplay
667 between AI and neuroscience research. *Neural Networks* **144**, 603-613 (2021).
- 668 6 Mathis, M. W. & Mathis, A. Deep learning tools for the measurement of animal behavior in
669 neuroscience. *Current opinion in neurobiology* **60**, 1-11 (2020).
- 670 7 Jhuang, H. *et al.* Automated home-cage behavioural phenotyping of mice. *Nat Commun* **1**,
671 68, doi:10.1038/ncomms1064 (2010).
- 672 8 Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive
673 machine learning for automatic annotation of animal behavior. *Nat Methods* **10**, 64-67,
674 doi:10.1038/nmeth.2281 (2013).
- 675 9 de Chaumont, F. *et al.* Real-time analysis of the behaviour of groups of mice via a depth-
676 sensing camera and machine learning. *Nature biomedical engineering* **3**, 930-942 (2019).

- 677 10 Gerós, A., Magalhães, A. & Aguiar, P. Improved 3D tracking and automated classification of
678 rodents' behavioral activity using depth-sensing cameras. *Behavior research methods* **52**,
679 2156-2167 (2020).
- 680 11 Lorbach, M., Poppe, R. & Veltkamp, R. C. Interactive rodent behavior annotation in video
681 using active learning. *Multimedia Tools and Applications* **78**, 19787-19806,
682 doi:10.1007/s11042-019-7169-4 (2019).
- 683 12 Marques, J. C., Lackner, S., Felix, R. & Orger, M. B. Structure of the Zebrafish Locomotor
684 Repertoire Revealed with Unsupervised Behavioral Clustering. *Curr Biol* **28**, 181-195 e185,
685 doi:10.1016/j.cub.2017.12.002 (2018).
- 686 13 Wiltschko, A. B. *et al.* Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121-
687 1135 (2015).
- 688 14 Geuther, B. Q. *et al.* Robust mouse tracking in complex environments using neural networks.
689 *Communications Biology* **2**, 124, doi:10.1038/s42003-019-0362-1 (2019).
- 690 15 Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with
691 deep learning. *Nat Neurosci* **21**, 1281-1289, doi:10.1038/s41593-018-0209-y (2018).
- 692 16 Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J. H. & de Polavieja, G. G.
693 idtracker.ai: tracking all individuals in small or large collectives of unmarked animals. *Nat*
694 *Methods* **16**, 179-182, doi:10.1038/s41592-018-0295-5 (2019).
- 695 17 Forys, B. J., Xiao, D., Gupta, P. & Murphy, T. H. Real-time selective markerless tracking of
696 forepaws of head fixed mice using deep neural networks. *Eneuro* **7** (2020).
- 697 18 Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nat Methods*
698 **16**, 117-125, doi:10.1038/s41592-018-0234-5 (2019).
- 699 19 Graving, J. M. *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose
700 estimation using deep learning. *Elife* **8**, e47994 (2019).

- 701 20 Dunn, T. W. *et al.* Geometric deep learning enables 3D kinematic profiling across species
702 and environments. *Nature methods* **18**, 564-573 (2021).
- 703 21 Bohoslav, J. P. *et al.* DeepEthogram, a machine learning pipeline for supervised behavior
704 classification from raw pixels. *Elife* **10**, e63377 (2021).
- 705 22 Marks, M. *et al.* SIPEC: the deep-learning Swiss knife for behavioral data analysis. *bioRxiv*
706 (2020).
- 707 23 Jiang, Z., Chazot, P. L., Celebi, M. E., Crookes, D. & Jiang, R. Social behavioral phenotyping of
708 *Drosophila* with a 2D–3D hybrid CNN framework. *IEEE Access* **7**, 67972-67982 (2019).
- 709 24 Nourizonoz, A. *et al.* EthoLoop: automated closed-loop neuroethology in naturalistic
710 environments. *Nature Methods* **17**, 1052-1059 (2020).
- 711 25 Kane, G. A., Lopes, G., Saunders, J. L., Mathis, A. & Mathis, M. W. Real-time, low-latency
712 closed-loop feedback using markerless posture tracking. *Elife* **9**, e61909 (2020).
- 713 26 Schweihoff, J. F. *et al.* DeepLabStream enables closed-loop behavioral experiments using
714 deep learning-based markerless, real-time posture detection. *Communications biology* **4**, 1-
715 11 (2021).
- 716 27 Sehara, K., Zimmer-Harwood, P., Larkum, M. E. & Sachdev, R. N. Real-time closed-loop
717 feedback in behavioral time scales using DeepLabCut. *Eneuro* **8** (2021).
- 718 28 Feichtenhofer, C., Fan, H., Malik, J. & He, K. in *Proceedings of the IEEE/CVF international*
719 *conference on computer vision*. 6202-6211.
- 720 29 Elboushaki, A., Hannane, R., Afdel, K. & Koutti, L. MultiD-CNN: A multi-dimensional feature
721 learning approach based on deep convolutional networks for gesture recognition in RGB-D
722 image sequences. *Expert Systems with Applications* **139**, 112829 (2020).
- 723 30 Zhang, L. *et al.* in *Proceedings of the IEEE International Conference on Computer Vision*
724 *Workshops*. 3120-3128.

- 725 31 Singh, R., Khurana, R., Kushwaha, A. K. S. & Srivastava, R. Combining CNN streams of
726 dynamic image and depth data for action recognition. *Multimedia Systems*, 1-10 (2020).
- 727 32 Machado, A. S., Darmohray, D. M., Fayad, J., Marques, H. G. & Carey, M. R. A quantitative
728 framework for whole-body coordination reveals specific deficits in freely walking ataxic
729 mice. *Elife* **4**, doi:10.7554/eLife.07892 (2015).
- 730 33 Ohayon, S., Avni, O., Taylor, A. L., Perona, P. & Roian Egnor, S. E. Automated multi-day
731 tracking of marked mice for the analysis of social behaviour. *J Neurosci Methods* **219**, 10-19,
732 doi:10.1016/j.jneumeth.2013.05.013 (2013).
- 733 34 Perez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & de Polavieja, G. G. idTracker:
734 tracking individuals in a group by automatic identification of unmarked animals. *Nat*
735 *Methods* **11**, 743-748, doi:10.1038/nmeth.2994 (2014).
- 736 35 Hong, W. *et al.* Automated measurement of mouse social behaviors using depth sensing,
737 video tracking, and machine learning. *Proc Natl Acad Sci U S A* **112**, E5351-5360,
738 doi:10.1073/pnas.1515982112 (2015).
- 739 36 Unger, J. *et al.* An unsupervised learning approach for tracking mice in an enclosed area.
740 *BMC Bioinformatics* **18**, 272, doi:10.1186/s12859-017-1681-1 (2017).
- 741 37 Simonyan, K. & Zisserman, A. in *Advances in neural information processing systems*. 568-
742 576.
- 743 38 Sturman, O. *et al.* Deep learning-based behavioral analysis reaches human accuracy and is
744 capable of outperforming commercial solutions. *Neuropsychopharmacology* **45**, 1942-1952
745 (2020).
- 746 39 Bohoslav, J. P. *et al.* DeepEthogram: a machine learning pipeline for supervised behavior
747 classification from raw pixels. *bioRxiv* (2020).

748 40 Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & De Polavieja, G. G. idTracker:
749 tracking individuals in a group by automatic identification of unmarked animals. *Nature*
750 *methods* **11**, 743-748 (2014).

751 41 Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M. & Burgard, W. in *2015 IEEE/RSJ*
752 *International Conference on Intelligent Robots and Systems (IROS)*. 681-687 (IEEE).

753 42 Madai-Tahy, L., Otte, S., Hanten, R. & Zell, A. in *International Conference on Artificial Neural*
754 *Networks*. 29-37 (Springer).

755

756 Acknowledgments

757 This work was partially financed by FEDER—Fundo Europeu de Desenvolvimento Regional funds through the
758 COMPETE 2020—Operacional Programme for Competitiveness and Internationalisation (POCI), Portugal
759 2020, and by Portuguese funds through FCT—Fundação para a Ciência e a Tecnologia/Ministério da Ciência,
760 Tecnologia e Ensino Superior in the framework of the projects PTDC/EMD-EMD/31540/2017 (POCI-01-0145-
761 FEDER-031540). We acknowledge the support of the i3S Animal House facility. Ana Gerós was funded by FCT
762 – Fundação para a Ciência e a Tecnologia, grant contract SFRH/BD/137385/2018.

763 Author contributions

764 Ana Gerós: Methodology, Software, Validation, Formal analysis, Visualization, Writing – original
765 draft preparation, Writing – review & editing; Ricardo Cruz: Software, Validation, Formal analysis,
766 Visualization, Writing – review & editing; Fabrice de Chaumont: Writing – review & editing; Jaime
767 S. Cardoso: Methodology, Writing – review & editing; Paulo Aguiar: Conceptualization,
768 Methodology, Writing – review & editing, Supervision.

769 Competing interests

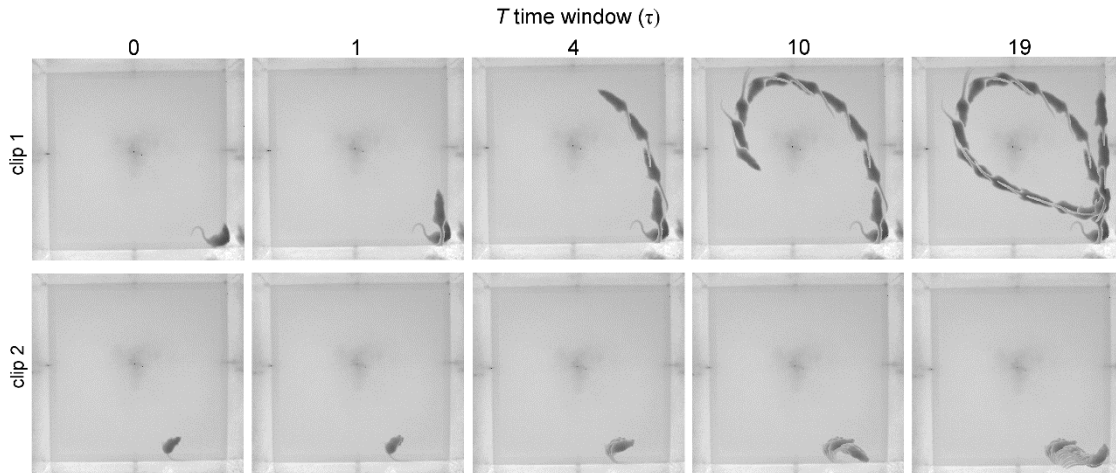
770 The authors declare no competing interests.

771 Ethics

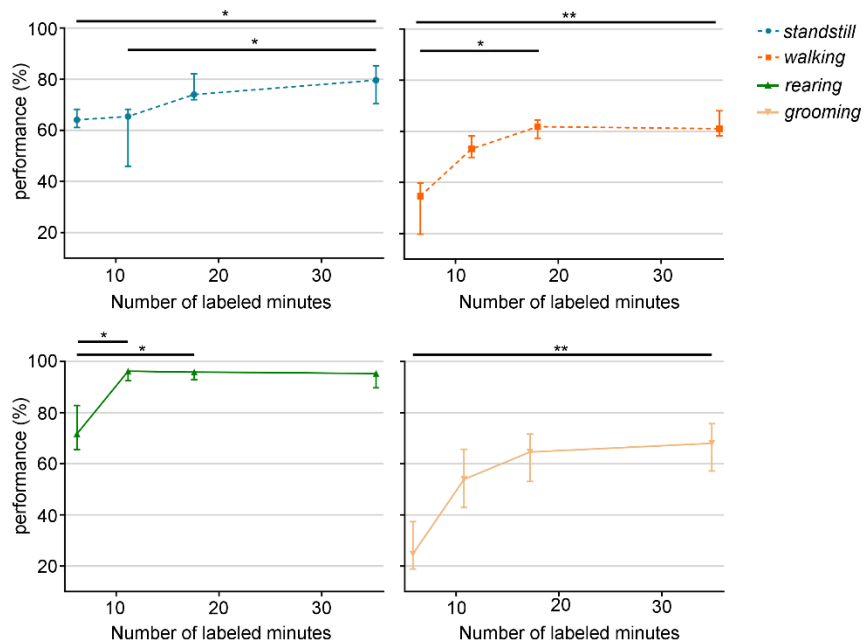
772 Animal experimentation: Animal housing and experimental procedures performed according to Portuguese
773 Legislation Dec. Lei nº113/2013 and the European Directive 2010/63/EU on the protection of animals used
774 for scientific purposes. The study was approved by 'Direção Geral de Alimentação e Veterinária' (Lisbon,
775 Portugal).

Supplementary Information

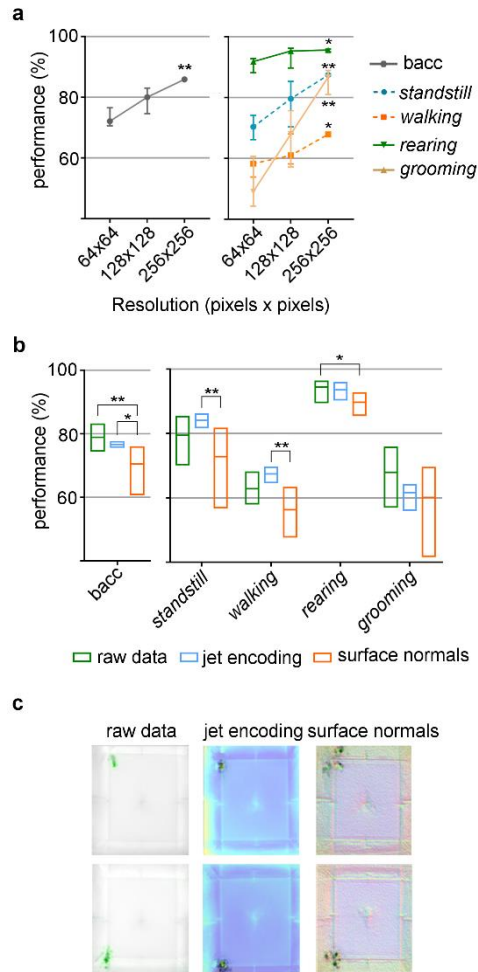
Supplementary files



Supplementary Figure 1. How much temporal information does the network need for rodents' behavioral learning? Stroboscopic montages in which each animal position represents raw depth frames extracted at every 133 ms, for 2 different walking clips and different time windows T , in units of τ ($\tau = 133$ ms). Each stroboscopic image illustrates the depth video sequence input fed to the deep learning network for different values of T .

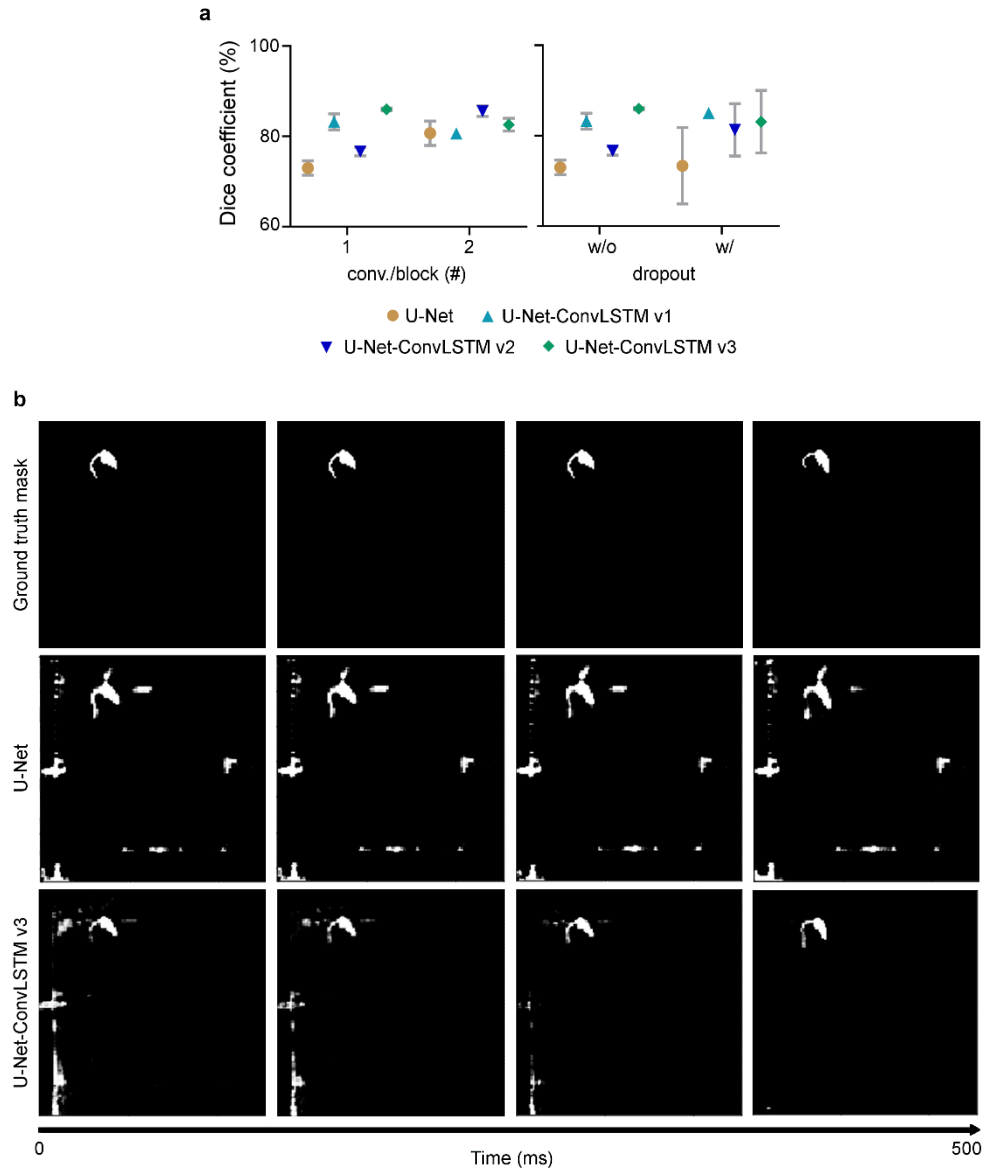


Supplementary Figure 2. How much information does the network need to learn? Extended statistical analysis for per-class classification performance as function of number of labeled minutes. Data represented as median \pm 95% confidence interval ($N = 5$ trials). * $p < 0.05$; ** $p < 0.01$.



Supplementary Figure 3. Which input sequence representation is most informative for network's learning? a.

Recognition performance of the single-branch architecture with different input resolutions. * and ** denote statistical significance when compared to the lowest resolution (64x64). **b.** Different depth encodings and corresponding performance, when compared to raw depth input frames. Data represented as median \pm 95% confidence interval (N = 5 trials). * $p < 0.05$; ** $p < 0.01$. **c.** Sensitivity analysis for different depth encoding methods (two different frames are shown), with gradients in green or black.



Supplementary Figure 4. Semantic segmentation results of U-Net-based networks. a. Networks' performance regarding Dice coefficient for different architectural parameters. Left: number of convolutional layers per block; Right: networks without (w/o) and with (w/) dropout layer at the end of the encoder. The traditional U-Net architecture was extended by placing a convolutional Long Short-Term Memory (ConvLSTM) layer at different positions in the network (U-Net-ConvLSTM), in order to find which position is most suitable for the depth images segmentation task (following Pfeuffer, et al. ¹ methodology). U-Net-ConvLSTM version 1 (v1) – ConvLSTM layer placed between the encoder and the decoder. U-Net-ConvLSTM version 2 (v2) – ConvLSTM layer placed in the end of the network. U-Net-ConvLSTM version 3 (v3) – a combination of the last two versions. Data represented as median \pm 95% confidence interval (N = 2 trials). **b.** Sample clips representing original (top) and predicted segmentation masks by the U-Net (middle) and U-Net-ConvLSTM v3 (bottom) networks, for a time window of 500 ms. Black pixels represent the background predictions and white pixels represent foreground (animal) predictions. During the inference, the presence of ConvLSTM layers improves the segmentation masks over time.

Additional Results

Input resolution improves behavioral classification performance

As part of the networks' study, the effect of input resolution was also examined, keeping the single-branch architecture with default parameters (**Supplementary Figure 3a**). As expected, the highest resolution (256x256) achieved the best results, with an overall performance of 85.9% [82.8 – 86.6]%. All behavioral events seem to benefit from increased resolution, in particular *grooming*, with an increase of approximately 44% over the lowest resolution. The fact that *grooming* events seem to need both higher temporal and spatial resolutions makes it the most sensitive and complex behavior to recognize.

Raw depth video inputs are the most informative for the learning process

Depth data encodes distance from the sensor to the captured scene and the information of each pixel is of a different nature, when compared to RGB images which were originally directly used as input for the CNNs. Thereby, the questions that arise are will CNNs learn as effectively when using raw depth images without any encoding, and, if not, how should a depth image be encoded to be used as inputs in CNNs so that it can learn more meaningful features for rodents' classification challenge. Networks were then trained with varying input depth encoding (**Supplementary Figure 3b**). Regarding per-class recognition, the negative effect on network's learning when using surface normal encoding is even more pronounced. One possible explanation is that when using a colorization method based on the calculation of surface normal, the reflexes on the walls of the open-field during, for example, grooming events (which are always near open-field's periphery) are more visible and may be interfering with networks' learning. Sensitivity analysis can be used to identify the most relevant input features during the learning process, by calculating heatmaps from pixel-wise normalized gradients (derivative of class model's predictions with respect to pixel values). This impact on model's prediction is exemplified on **Supplementary Figure 3c**, where, by

using surface normals, periphery pixels seem to have a stronger influence on model's prediction (gradient colored as black pixels), when compared to pixels from networks trained with raw depth frames (gradient colored as green pixels). Overall, behavioral learning does not seem to benefit from any of these typical depth input representations.

References

- 1 Pfeuffer, A., Schulz, K. & Dietmayer, K. in *2019 IEEE Intelligent Vehicles Symposium (IV)*. 1441-1447 (IEEE).