A Workflow for Developing Biohybrid Intelligent Sensing Systems

Edoardo Fazzari[1,2,*], Fabio Carrara[3], Fabrizio Falchi[3], Cesare Stefanini[1,2] and Donato Romano[1,2]

[1]The Biorobotics Institute, Viale Rinaldo Piaggio, Pontedera, 56025, Italy
[2]Department of Excellence in Robotics and AI, Sant'Anna School of Advanced Studies, Piazza Martiri della Libertà, Pisa, 56127, Italy
[3]Institute of Information Science and Technologies of the National Research Council of Italy (ISTI-CNR), via G. Moruzzi, Pisa, 56124, Italy

Abstract

Animal are sometime exploited as biosensors for assessing the presence of volatile organic compounds (VOCs) in the environment by interpreting their stereotyped behavioral responses. However, current approaches are based on direct human observation to assess the changes in animal behaviors associated to specific environmental stimuli. We propose a general workflow based on artificial intelligence that use pose estimation and sequence classification technique to automate this process. This study also provides an example of its application studying the antennae movement of an insect (e.g. a cricket) in response to the presence of two chemical stimuli.

Keywords

biosensor, deep learning, pose estimation, sequence classification, cricket, biohybrid system

1. Introduction

Animal biosensors are analytical tools that exploit animal olfactory capabilities to identify volatile organic compounds (VOCs) [1], through the conversion of biological selective responses into measurable signals. This approach is important for three reasons: a) Animals sensitive olfactory system is beyond human capabilities and electronic devices; b) Biosensors can be portable, easy-to-use, eco-friendly, and do not need any manufacturing process for the analysis; c) Biosensors have potential application in a wide range of fields, from ecological studies to biomedical uses.

Previous research has primarily focused on the detection of explosives and narcotics [2, 3], medical diagnosis [4], and the use of animal biosensors as early warning systems for forest fires [5]. This paper focus on classifying the type of response of crickets postexposure to two chemical substances, namely ammonia and sucrose powders, through the analysis of the movement of their antennae. Furthermore, while previous studies have relied on user inputs or direct nerve stimuli readings [6], our work emphasizes the development of an autonomous and intelligent workflow utilizing computer vision techniques.

Recent advancements in animal pose estimation neural networks, such as SLEAP [7], have demonstrated state-of-the-art performance in detecting keypoints on animals. These techniques can be applied to track animal movements and generate time sequences that are useful for the study of animal behavior. Such applications have the potential to improve the performance of animal biosensors.

The contribution of this work is the development of a workflow based on pose estimation and sequence processing techniques (extendable to diverse applications) for developing *Biohybrid Intelligent Sensing Systems* (BISS) that are sustainable, and do not require the installation of any device on the animal (ethically acceptable and eco-friendly). We investigated how this workflow is effective for the aforementioned task, otherwise impossible for a human user. The development of AI architectures can help to reduce some limitation of the use of animal biosensor, such as errors related to human observation and interpretation, as well as increasing the method calibration and standardisation.

2. Materials and Methods

We here describe the models, dataset and proposed workflow.

2.1. Models

We base our pose estimation on SLEAP configuring it with an UNet[8] backbone. We tested multiple configurations of hyper-parameters through grid search to find

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Corresponding author.
edoardo.fazzari@santannapisa.it (E. Fazzari); fabio.carrara@isti.cnr.it (F. Carrara); fabrizio.falchi@cnr.it (F. Falchi); cesare.stefanini@santannapisa.it (C. Stefanini); donato.romano@santannapisa.it (D. Romano)
0000-0002-4570-4170 (E. Fazzari); 0000-0001-5014-5089 (F. Carrara); 0000-0001-6258-5313 (F. Falchi); 0000-0003-0989-041X (C. Stefanini); 0000-0003-4975-3495 (D. Romano)

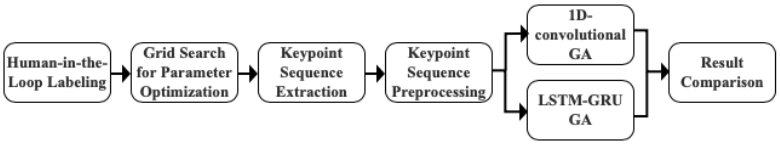

Figure 1: Main steps undertaken for the development of a *Biohybrid Intelligent Sensing Systems* (BISS) for relating animal movements to specific environment stimuli.

the best combination of max stride, filters number and input scaling.

For the classification network, we assessed the performance of LSTM-GRU[9, 10] and 1D-convolutional[11] neural networks, searching the best architecture via genetic algorithm.

2.2. Dataset

For this study, adult crickets (*Acheta Domesticus*) were obtained from an e-commerce site and maintained in controlled conditions. A total of 69 crickets were selected based on size and antennae visibility. To ensure no behavioral bias, only one video was taken per cricket. Crickets were placed in a Petri dish with one of three stimuli: nothing (i.e., control case), sucrose, or ammonia powder. Each recording was longer than 3 minutes and comprised two parts, the "settling in" (1 minutes) and "interaction" (2 minutes) periods. An iPhone 14 Pro was used to record the Petri dish and a light panel was used to reduce reflections. The resulting dataset is balanced and includes 23 videos for each stimulus, totaling 3 hours, 37 minutes, and 56 seconds.

2.2.1. Dataset Processing

The obtained videos were preprocessed by reducing the frames per second (fps) to 29 to ensure equal measurement across all videos. The "interaction period" was identified between frames 1740 and 5220, resulting in 3480 frames. The videos containing only the interactive part were reformatted to 1080x1080 pixels, centering the Petri dish and removing pixels from outside the Petri dish that could interfere with the neural network's learning for pose estimation. This was done to ensure consistency in the dataset, enabling unbiased and accurate data analysis.

2.3. Proposed Workflow

We describe here the suggested workflow steps for the development of BISS referring only to the techniques exploited and not to our specific experiments (see section 3 for that instead). Figure 1 schematizes the task undergone in our workflow.

Human-in-the-Loop Labeling To generate the labels required to train our SLEAP model, we adopted the human-in-the-loop approach developed by Pereira et al. [12]. This method involves labeling a restricted number of frames, training the pose estimation model, and then using it to produce new labels for unlabeled frames. Additionally, incorrectly placed keypoints are repositioned, and the new labels are combined with the previously labeled ones to retrain the model from scratch. A fixed validation set is used to determine when the process should be terminated, and this is done by comparing the results in terms of mean Average Precision (mAP) between each training-labeling iteration. If there is no improvement in mAP despite increasing the number of labeled frames, then the labeling phase is concluded.

Grid Search for Parameter Optimization Once a suitable training set has been identified in the previous step, the subsequent task is to optimize the pose estimation architecture by modifying the parameters that have the most significant impact, such as max stride, initial number of filters, and input scaling. To limit the number of training runs, it is crucial to consider the configuration used in the previous phase and follow three key guidelines: Firstly, if the objective is to identify fine features characterized by a small number of pixels, it is recommended to increase the value of input scaling. Conversely, if most of the keypoints are already detected, it is advisable to reduce the value of input scaling to obtain a smaller model. Secondly, to ensure that the entire animal is covered, the receptive field should be resized by changing the value of the max stride. Lastly, the initial number of filters should be tested with values of 32 and 64, and the preference should be given to the lower value, i.e., a smaller and faster network, even if it results in the same mAP.

Keypoint Sequence Extraction and Preprocessing Identified the best pose estimation model, tracking sequences can be obtained for all videos in the dataset. Before proceeding to the next stage, each sequence should undergo preprocessing, which includes implementing a filling strategy to remove any NaN values, and normalizing the values by subtracting the mean and dividing by the standard deviation. In our workflow, the filling strat-

egy is tailored to each specific keypoint sequence and utilizes the following formula to fill in a missing values in position t:

$$v_t = \frac{\alpha v_{t+k} + v_{t-1}}{1+\alpha}, \quad \alpha = 1/k \tag{1}$$

where k+t is the first subsequent frame with a non-NaN value for the key-point which we are considering. An important case was also handled when the value for t=0 is NaN, in this case the value is set to the first subsequent non-NaN value.

Genetic Algorithm (GA) for Architecture Development In this stage, the search for the best classification model is undertaken and constructed using a Genetic Algorithm (GA). This method typically results in the development of models with good predictive accuracy at a relatively low cost compared to other approaches, such as random search [13]. In order to employ a GA, six key parameters are taken into consideration: the initial population, fitness function, selection, crossover, and mutation functions, as well as the chromosome structure. The size of the initial population should be chosen based on the computational power required, as a larger population will take longer to compute due to increased training, while a smaller one may lead to inferior results. The fitness function, which is developed as a maximization objective, is defined as follows:

$$\text{fit(gene)} = \begin{cases} -10 \cdot (1 - \text{train\_accuracy}) & \text{if } a \\ -15 & \text{if } b \\ -20 & \text{if } c \\ -\text{val\_loss} & O/W \end{cases} \tag{2}$$

where a stands for "if the training or validation accuracies are less or equal than 1 over the number of classes, or the training accuracy is less than the validation accuracy"; b stands for "if the training accuracy is less than 0.1"; c stands for "no convolutional or RNN layers are present". The decision to create such fitness function, rather than simply minimizing the validation loss, is motivated by the fact that for problems with limited data and inherent complexity, such in our experiment, it is possible for a model to achieve a validation loss that is close to, or even lower than, models with better validation accuracy and above the random guess. To overcome this issue, the finest function was designed to take into account the training accuracy, enabling another metric for evaluating the network's quality. Higher training accuracy values lead to fitness values closer to those obtained from the validation loss, indicating a kind of network goodness that can be utilized in subsequent GA iterations. Additionally, the "train_accuracy<val_accuracy" check is incorporated to prevent the genetic algorithm from overfitting on the validation accuracy, which could adversely affect the training accuracy and hinder the ability to generalize effectively. Finally, we check if the value of the training accuracy is less than 0.1, setting a default value for this case. This was done to prevent a false suggestion of good models using the first case in Equation 2.3.

The employed selection algorithm is tournament selection, which randomly selects a predetermined number of individuals from the population and then selects the most fit individual from the group, adding it to the mating pool. Along with tournament selection, elitism was incorporated as a selection strategy. The chosen crossover algorithm was bounded simulated binary crossover (SBX), which is a bounded version of the Simulated Binary Crossover (SBX)[14]. Lastly, the mutation function utilized was bounded polynomial mutation, which is a bounded mutation operator that uses a polynomial function for the probability distribution.

The chromosomes used to construct the one-convolutional network are composed of 56 real-coded genes. The first block consists of six genes repeated 5 times indicating: 1) the presence of the convolutional block (0 if absent, 1 if present); 2) the number of filters of the one-convolutional layer (16 to 1024); 3) presence of the batch normalization layer (0 if absent, 1 if present); 4) activation function (0: sigmoid, 1: swish, 2: tanh, 3: relu, 4: gelu, 5: elu, 6: leaky relu); 5) presence of dropout (0 if absent, 1 if present); 6) dropout rate (0 to 0. 5, considering only multiple values of 0.05). Following this, a gene is used to indicate the type of connection between convolutional and fully connected layers. This gene serves to determine wheter the layer is Flatten or GlobalAveragePooling1D (0 indicating the former, and 1 the latter). The second block consists of 5 genes indicating: 1) the presence of the fully connected block (0 if absent, 1 if present); 2) the number of units (3 to 512); 3) activation function; 4) presence of dropout; 5) dropout rate. The chromosomes used to construct the RNN are composed of 50 real-coding genes instead. Compared with the previous case, the only changes are related to the first block and the removal of the gene related to the connection between the convolutional and the fully connected part, which is not necessary in this case. The first block consists of 5 genes repeated 5 times indicating: 1) the presence of the RNN block; 2) the use of a bidirectional layer (0 if absent, 1 if present); 3) type of RNN (0: LSTM, 1: GRU); 4) number of units (16 to 1024); 5) activation function.

Compare GA Results After finding the best model using the genetic algorithm for both the convolutional and RNN cases, the models were evaluated using iterated K-fold validation. This validation technique involves randomly shuffling the dataset and splitting it into training

and validation sets, then running K-fold validation multiple times. This method is crucial in cases where the available data is limited, as in our study, and an accurate evaluation of the model is needed. The final score is obtained by calculating the mean accuracy across all K-fold validation runs.

3. Results & Discussion

In this section, we present the findings of our experiments. Firstly, we elaborate on the results derived from human-in-the-loop testing, followed by a detailed exposition of the optimal configuration parameters that were identified for the pose estimation model in subsection 3.1. Subsequently, in subsection 3.2, we showcase the two models (namely LSTM-GRU and convolutional architectures) that were discovered by the genetic algorithm in tandem with a comparative analysis of their performance based on the tracking sequence obtained from our SLEAP model. Finally, each section entails a discussion of the limitations inherent in our methodology, along with a discourse on possible future extensions that may serve to enhance the effectiveness of our workflow.

3.1. Pose Estimation

Our experiment focused on determining whether it was feasible to predict the chemical composition of substances (i.e., sucrose or ammonia powders) by analyzing the movement of a cricket's antennae. In order to achieve this, we strategically placed five keypoints at the proximal and distal ends of the antennae (both left and right) as well as one over the head.

To generate the required labels for training the SLEAP model, a human-in-the-loop approach was utilized. The SLEAP model employed had a maximum stride value of 64, an initial filter rate of 64, and input scaling of 0.7. This process produced a total of 5460 training frames and resulted in a mean average precision (mAP) value of 0.804768, which was validated using 300 frames from videos different from those used in the training set.

A grid search was performed on the obtained training set by varying the maximum stride value (32 or 64), initial filter rate (32 or 64), and input scaling (increased by 0.1 until it reached 1.0). The optimal parameter configuration was determined to be a maximum stride value of 64 and an input scaling of 1.0, which achieved an mAP of 0.837392 on the validation set. These parameters were selected based on the guidelines outlined in subsection 2.3, as detecting fine features such as the distal end of the antennae (characterized by 4 ± 1 pixels in width) required the parameters to be set to their maximum suggested values.

The mean average precision (mAP) value obtained is strongly associated with the difficulty of locating the distal ends of the antennae, particularly when the crickets are situated close to the wall of the Petri dish. In such instances, the two antennae are more prone to overlap, as is evident in Figure 2. The occurrence of such errors can be mitigated by taking into account temporal information, such as previous and subsequent frames, to predict for a single frame. Although such architecture already exists in the Animal Pose Estimation (APE) domain[15], it employs a less complex neural network[12] compared to the one utilized in this study. Although this may yield higher mAP values, the computational cost may increase substantially.

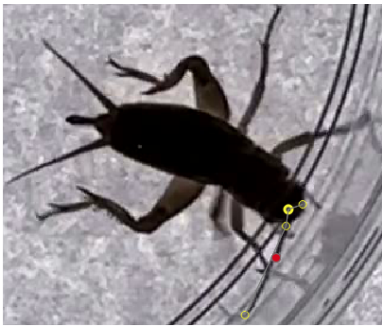

Figure 2: The image depicts an instance in which our model has erroneously labeled a frame by misplacing the right and left distal ends in a single location. A correction of the placement of the right distal end is indicated by a red dot, which was achieved by scrutinizing the temporal information. This particular example serves to underscore the prospective advantages that may be derived from integrating temporal context into the keypoint detection process.

Training details: we train each model for 400 epochs with batch size of 8 using Adam optimizer. The initial learning rate is set as 1e-4 and we made use of the SLEAP's default learning rate decay strategy, with a patience of 20 epochs and a minimum delta of 1e-8. To mitigate the potential risk of overfitting, we incorporated the early stopping technique, terminating the training when the validation loss did not decrease for 50 epochs.

3.2. Chemical Interaction Classification

The genetic algorithm hyperparaters for the population, the number of epochs and the elitism were set to 250, 20 and 10 respectively. The structure of the best generated CNN consist of a convolutional layer with 821 filters, followed by a batch normalization layer and a tanh activation layer. This was followed by a convolutional layer with 821 filters and an elu activation layer, which was in turn followed by a dropout layer with a rate of 0.2. The final convolutional layer contained 483 filters and a tanh activation layer. The output of the convolutional

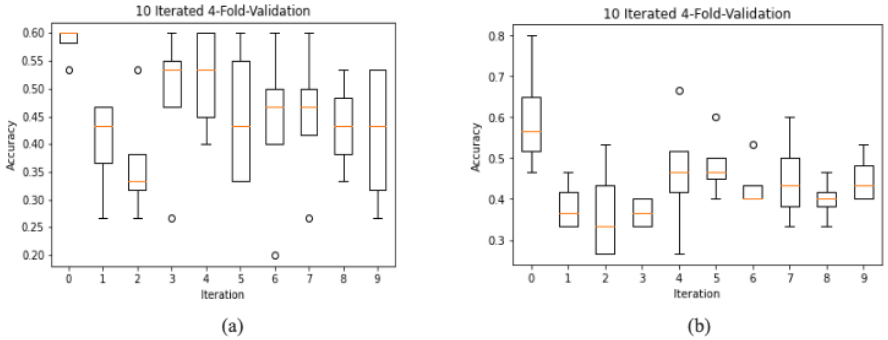

Figure 3: Boxplots for the 10 iterated 4-fold validation of the best models generated by the genetic algorithm. (a) depicts the boxplot related to the convolutional architecture; (b) for the recurrent neural network.

layers was flattened. The RNN was structured as a bidirectional LSTM layer with 707 units and an elu activation function, followed by a GRU layer with 660 units and a leaky relu activation function. This was followed by a bidirectional GRU layer with 469 units and a leaky relu activation function, a dense layer with 138 units and a gelu activation function and a dropout layer with a rate of 0.2. This was in turn followed by a dense layer with 150 units and a leaky relu activation function. The validation accuracy obtained by the two models was 58.33% and 50% respectively.

The effective comparison between the two models was conducted using 10 iterated 4-fold validation, and the corresponding boxplot for each iteration is presented in Figure 3. Notably, the recurrent neural network exhibited superior accuracy in the initial iteration, achieving a noteworthy 80% accuracy. However, upon averaging the results, it performed slightly worse compared to the convolutional neural network. Specifically, the average accuracy was 45.33% ±5.85% for the former and 44% ±6.6% for the latter, with the generated-CNN model proving marginally more effective. It is essential to acknowledge that while the outcomes are not extraordinary, they are still superior to the current human capabilities, as discerning antennae movement remains a challenging task. Additionally, the results may be attributed to the limited attention span of animals and the potential impact of behavioral variations[1]. The crickets used in the study were not trained to exhibit specific behaviors towards the two powders; rather, their innate behavior was assessed. Moreover, it is worth noting that strategies to train crickets to respond in a specific manner to certain stimuli already exist[16, 17] and can be introduced to our workflow to further enhance the accuracy of our models. Therefore, future studies may incorporate such training methodologies to overcome the potential limitations of using untrained crickets as sensors. By doing so, we can not only improve the performance of the models but also pave the way for the development of novel sensor technologies that are inspired by natural systems.

Training details: we trained each model constructed for the genetic algorithm and the iterated K-fold validation using 1000 epochs with a batch size of 16. To prevent from overfitting, we made use of early stopping, terminated the training when the validation loss did not decrease for 100 epochs. The learning rate was reduced on plateau with a minimum delta of 1e-3 and a patience of 50.

4. Conclusion

This paper proposes a novel workflow for the creation of Biohybrid Intelligent Sensing Systems (BISS) utilizing deep learning techniques, specifically those pertaining to convolutional and recurrent neural networks. The underlying motivation for this approach is to enhance the performance and broaden the spectrum of potential applications of animal biosensors, by facilitating a more precise mapping of animal behaviors to the identification of volatile organic compounds or other environmental changes. The development of such methodologies has the potential to address certain limitations associated with the use of animal biosensors, such as the introduction of errors stemming from human observation and interpretation, while also facilitating method standardization. Additionally, by relying solely on recordings, BISS presents an ethical and environmentally sustainable alternative.

In our upcoming endeavors, we plan to incorporate temporal information in our pose estimation task to further enhance our workflow. Furthermore, we intend to perform experiments with trained animals to more accurately gauge how our operations can significantly improve the current state-of-the-art in animal biosensors.

References

[1] Y. Oh, Y. Lee, J. Heath, M. Kim, Applications of animal biosensors: A review, Sensors Journal, IEEE 15 (2015) 637–645. doi:10.1109/JSEN.2014.2358261.

[2] D. Olson, G. Rains, Use of a parasitic wasp as a biosensor, Biosensors 4 (2014) 150–160. URL: https://www.mdpi.com/2079-6374/4/2/150.

[3] K. Taylor-McCabe, R. M. Wingo, T. K. Haarmann, Honey bees (apis mellifera) as explosives detectors: exploring proboscis extension reflex conditioned response to trinitrotolulene (tnt), Apidologie (2008).

[4] D. H. Pickel, G. P. Manucy, D. B. Walker, S. B. Hall, J. C. Walker, Evidence for canine olfactory detection of melanoma, Applied Animal Behaviour Science 89 (2004) 107–116.

[5] S. D. H. Permana, G. Saputra, B. Arifitama, Yaddarabullah, W. Caesarendra, R. Rahim, Classification of bird sounds as an early warning method of forest fires using convolutional neural network (cnn) algorithm, Journal of King Saud University - Computer and Information Sciences 34 (2022) 4345–4357. URL: https://www.sciencedirect.com/science/article/pii/S1319157821000999. doi:https://doi.org/10.1016/j.jksuci.2021.04.013.

[6] D. Saha, D. Mehta, E. Altan, R. Chandak, M. Traner, R. Lo, P. Gupta, S. Singamaneni, S. Chakrabartty, B. Raman, Explosive sensing with insect-based biorobots, Biosensors and Bioelectronics: X 6 (2020) 100050. URL: https://www.sciencedirect.com/science/article/pii/S2590137020300169. doi:https://doi.org/10.1016/j.biosx.2020.100050.

[7] T. Pereira, N. Tabris, A. Matsliah, D. Turner, J. Li, S. Ravindranath, E. Papadoyannis, E. Normand, D. Deutsch, Z. Wang, G. McKenzie-Smith, C. Mitelut, M. Castro, J. D'Uva, M. Kislin, D. Sanes, S. Kocher, S. Wang, A. Falkner, M. Murthy, Sleap: A deep learning system for multi-animal pose tracking, Nature Methods 19 (2022) 1–10. doi:10.1038/s41592-022-01426-1.

[8] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, CoRR abs/1505.04597 (2015). URL: http://arxiv.org/abs/1505.04597. arXiv:1505.04597.

[9] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735.

[10] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734. URL: https://aclanthology.org/D14-1179. doi:10.3115/v1/D14-1179.

[11] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751. URL: https://aclanthology.org/D14-1181. doi:10.3115/v1/D14-1181.

[12] T. Pereira, D. Aldarondo, L. Willmore, M. Kislin, S. Wang, M. Murthy, J. Shaevitz, Fast animal pose estimation using deep neural networks, Nature Methods 16 (2019) 117–125. doi:10.1038/s41592-018-0234-5.

[13] J. Rala Cordeiro, A. Raimundo, O. Postolache, P. Sebastião, Neural architecture search for 1d cnnsm-dash;different approaches tests and measurements, Sensors 21 (2021). URL: https://www.mdpi.com/1424-8220/21/23/7990. doi:10.3390/s21237990.

[14] R. Agrawal, K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (2000).

[15] H. Russello, R. van der Tol, G. Kootstra, T-leap: Occlusion-robust pose estimation of walking cows using temporal information, Computers and Electronics in Agriculture 192 (2022) 106559. URL: https://www.sciencedirect.com/science/article/pii/S0168169921005767. doi:https://doi.org/10.1016/j.compag.2021.106559.

[16] Y. Matsumoto, M. Mizunami, Context-dependent olfactory learning in an insect, Learning memory (Cold Spring Harbor, N.Y.) 11 (2004) 288–293. doi:10.1101/lm.72504.

[17] Y. Matsumoto, Learning and memory in the cricket gryllus bimaculatus, Physiological Entomology 47 (2022). doi:10.1111/phen.12387.

A. Online Resources

The code and the data are available via

- GitHub,
- Google Drive.