Using AI to Decode the Behavioral Responses of an Insect to Chemical Stimuli: Towards Machine-Animal Computational Technologies

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

[1]The BioRobotics Institute, Sant'Anna School of Advanced Studies, 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, National Research Council of Italy, via G. Moruzzi, Pisa, 56124, Italy.

*Corresponding author(s). E-mail(s): edoardo.fazzari@santannapisa.it; donato.romano@santannapisa.it;
Contributing authors: fabio.carrara@isti.cnr.it; fabrizio.falchi@cnr.it; cesare.stefanini@santannapisa.it;

Abstract

Orthoptera are insects with excellent olfactory sense abilities due to their antennae richly equipped with receptors. This makes them interesting model organisms to be used as biosensors for environmental and agricultural monitoring. Herein, we investigated if the house cricket *Acheta domesticus* can be used to detect different chemical cues by examining the movements of their antennae and attempting to identify specific antennal displays associated to different chemical cues exposed (e.g., sucrose or ammonia powder). A neural network based on state-of-the-art techniques (i.e., SLEAP) for pose estimation was built to identify the proximal and distal ends of the antennae. The network was optimised via grid search, resulting in a mean Average Precision (mAP) of 83.74%. To classify the stimulus type, another network was employed to take in a series of keypoint sequences, and output the stimulus classification. To find the best one-dimensional convolutional and recurrent neural networks, a genetic algorithm-based optimisation method was used. These networks were validated with iterated K-fold validation, obtaining an average accuracy of 45.33% for the former and 44% for the latter. Notably, we published and introduced the first dataset on cricket recordings that relate this animal's behaviour to chemical stimuli. Overall, this study proposes a novel and simple automated method that can be extended to other animals for the creation of Biohybrid Intelligent Sensing Systems (e.g., automated video-analysis of an organism's behaviour) to be exploited in various ecological scenarios.

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

1 Introduction

In recent years, Artificial Intelligence (AI) has become a critical tool in various biological research fields, such as medicine (Litjens et al. (2017)), agriculture (Jha et al. (2019)) and environmental monitoring (Couzin and Heins (2023)). Deep Learning, a subset of AI, has been instrumental in surpassing human performance in complex, time-consuming tasks (Khanzode and Sarode (2020)). This has enabled the development of new precision techniques that contribute to improving environmental sustainability, and with significant socio-economic implications (Rolnick et al. (2019)).

The advent of such precision techniques through the application of Deep Learning has paved the way for the construction of Biohybrid Intelligent Sensing Systems (BISSs), which represents an innovative approach to animal biosensors that utilizes artificial intelligence to detect changes and analyze the environment. As a result, the integration of AI in biological research fields, such as medicine, agriculture, and environmental monitoring, has led to significant advancements that have the potential to enhance the performance of BISSs and ultimately lead to their automation and optimization, highlighting the critical importance of this research area.

The development of animal biosensor systems is of paramount significance if we aim at comprehending the environment in a sustainable sound manner (Romano et al. (2019)) due to the numerous advantages they offer in comparison to traditional analytical tools. These advantages include portability, rapidity, ease of use, and cost-effectiveness without the need for a manufacturing process, chemical or biological reagents, or any other processing for the analysis (Oh et al. (2015)). Biosensors take advantage of animals' remarkable olfactory capabilities in detecting molecules in the atmosphere, with applications in numerous fields such as medical diagnosis (Pickel et al. (2004)), explosives detection (Taylor-McCabe et al. (2008)) and narcotic detection (Olson and Rains (2014)), thus proving to be highly effective. The aforementioned studies needed an observer to monitor the behavior of the animal to make a diagnosis or detect any abnormalities. However, techniques that do not require user input are based on the direct reading of nerve stimuli from the animal, which is then subjected to analysis. As an example, Saha et al. (2020) examined the signals from olfactory receptor neurons in the locust *Schistocerca americana* to ascertain if this insect could detect the smell of explosives and its concentration.

In this context, Animal Pose Estimation (APE) techniques using AI could represent a further step towards automating and optimizing animal biosensor systems. Technologies such as DeepLabCut (Mathis et al. (2018)) and SLEAP (Pereira et al.

(2022)) enable the tracking of animal(s) movements and the generation of time sequences that are the key for the study of animal behavior. Furthermore, Luxem et al. (2022) improved the analysis of animals' behaviors proposing a deep variational embeddings-based model for identifying behavioral structures in animals, using the tracking sequences obtained from mice recordings without requiring supervised or a-priori human interference. Additionally, Fang et al. (2021) proposed an architecture for behavioral classification based on a naïve Bayesian model to identify chickens' behavior for diagnosing poultry diseases using the sequences obtained from the pose estimation. However, these approaches are limited to defining the vocabularies of motifs without providing any correlation with the animal's response to environmental stimuli.

To bridge this gap, this paper proposes a workflow based on pose estimation techniques to develop a BISS that can identify the type of response generated in the house cricket *Acheta domesticus L. (Orthoptera: Gryllidae)* by different chemical cues. Unlike the previously cited study, the approach we used is non-invasive as the proposed technique exploits machine learning to analyze video-recordings of the cricket's antennae movements.

House crickets possess significant mechanosensory and chemosensory organs (Loudon et al. (2014)). However, to the best of our knowledge, the automatic association between antennae movements and odor recognition in crickets has not been explored so far. Although studies on insect antenna movements related to odor recognition have been conducted (i.e., ants), none of them included artificial intelligence techniques (Draft et al. (2018)). The issue with ants or other small insects is that their antennae are thin in comparison to crickets, whose antennae are significantly more elongated and therefore more easily discernible to the human eye, which increases the likelihood of being detected by automated pose estimation techniques. Moreover, crickets' antennae have been demonstrated to be highly dynamic and responsive (Yamawaki and Ishibashi (2014)), allowing for a higher probability of acquiring dynamic tracking sequences than insects with shorter antennae and limited mobility.

This research develops a workflow based on artificial intelligence techniques that can identify the type of response exhibited by crickets when exposed to certain chemical stimuli by considering their antennae's movement. This initial approach is intended to lay the groundwork for the creation of Biohybrid Intelligent Sensing Systems that are fully autonomous, eco-friendly, and sustainable. These systems will use computer vision and sequence processing techniques and will only require cameras and a few other electronic components, thus having minimal impact on the environment. The main contributions of this work are three-fold:

1. We present a streamlined workflow for mapping animal movements to stimuli, offering a straightforward approach to understanding the relationship between the two.
2. We introduce a novel fitness function designed to construct neural networks in scenarios with limited samples and a high risk of overfitting on the validation set. This innovative approach addresses the challenge of achieving desirable results on the validation set while maintaining suboptimal performance on the training set.
3. To the best of our knowledge, we are the first to publish a dataset specifically focused on crickets, enabling precise pose estimation from high-resolution videos.

As an additional contribution, the code and dataset used are available online (see section 5).

2 Materials and Methods

2.1 Materials and Dataset

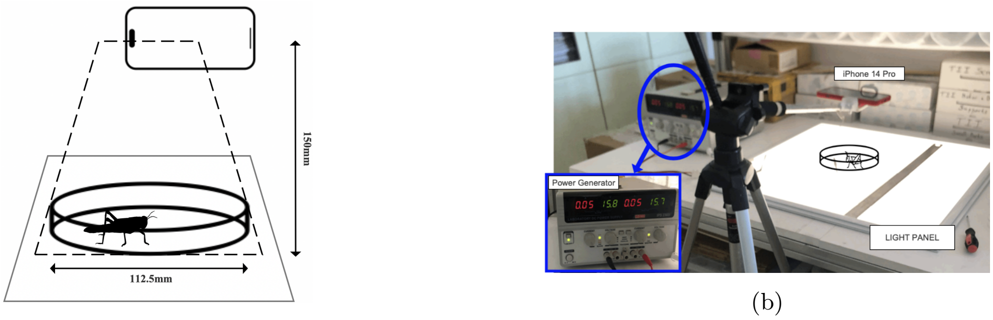
Adult crickets (*Acheta domesticus*) used in this study were obtained from an e-commerce site and maintained at the Institute of BioRobotics in Pontedera, Italy, at a temperature of 22 ± 1 °C, 55 ± 5% relative humidity, and 12:12h light:dark photoperiod. Of the 200 crickets purchased, 69 were selected for the experiment based on size (10-15 mm) and visible antennae (not injured), so that movement analysis could be conducted. To ensure that the crickets did not have a behavioral bias after interacting with a stimulus, only one video was taken for each cricket.

Individual crickets were placed in a closed Petri dish (112.5 mm in diameter) with one of the following three stimuli placed on a piece of paper: nothing (control case, C), solution of water and sucrose powder (sucrose case, S), and cake ammonia (ammonia case, A). 0.05 grams of each substance was used. 23 videos were recorded for each stimulus, resulting in a balanced dataset of 3 hours, 37 minutes, and 56 seconds. Each recording was longer than 3 minutes and comprised two parts: the first minute (from 0 to 59 seconds) was regarded as the "settling in" period, while the "interaction period" began from minutes 1 to 3. An iPhone 14 Pro was used to record the Petri dish, set to "Most Compatible" and 1080p at 30fps to demonstrate that professional video cameras are not necessary to obtain videos suitable for scientific experiments. To reduce the reflection of the smartphone on the Petri dish, a light panel placed in direct current at 0.11A and 16V was used. The smartphone was placed at a distance of about 150mm to capture the entire width of the Petri dish. Figure 1 shows the setup used to carry out the recordings. Despite the use of the light panel, a problem of reflection persisted when crickets were close to the wall of the Petri dish, resulting in a reflection on their legs and antennae on the wall creating a possible confusion for the pose estimation task. Figure 2(a) shows a cricket in a situation where the antennae make a reflection on the wall of the petri dish. Figure 2(b), Figure 2(c) and Figure 2(d) show crickets in the case when the stimulus is C, S, A, respectively.

To the best of our knowledge, this is the first public dataset containing crickets recordings.

2.2 Dataset Preprocessing

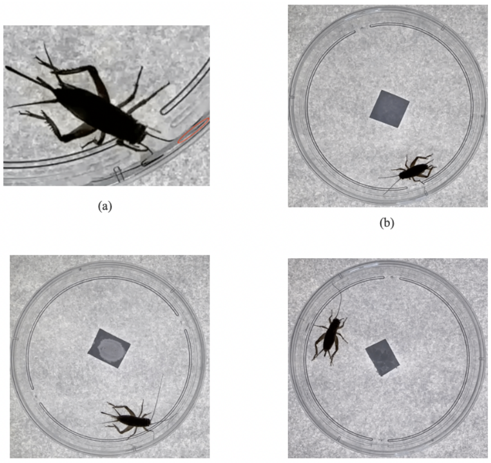
Prior to identifying the location of the antennae, a preprocessing step was performed on the recorded videos to extract the "interaction period." In order to ensure uniformity in measurement across all videos, the frame rate was reduced to 29 FPS, which was the lowest among all the videos. This crucial step allowed for the standardization of video length. The interaction period was then identified between frames 1740 (29x60) and 5220 (29x180), resulting in a total of 3480 frames per video. Furthermore, to center the petri dish and standardize its position in all videos, the videos were reformatted from 1920x1080 to 1080x1080. This resizing eliminated any extraneous pixels from outside



(a)

(b)

Fig. 1: (a) shows an illustration of the setup employed for recording the crickets. (b) shows the real setup in our lab.



(a)

(b)

(c)

(d)

Fig. 2: The image presented in (a) illustrates the scenario in which a reflection occurs between the cricket and the wall of the Petri dish. On the other hand, (b), (c), and (d) depict the visual representation of the stimuli C, S, and A, respectively.

the petri dish that may have had the potential to interfere with the neural network's learning process for pose estimation. All data processing was performed using Python on a MacOS Ventura 13.2 with 6-core i5 processors clocking at 3.9GHz and an iMac with (2*4) G RAM.

2.3 Cricket Pose Estimation

For each video analyzed, a pose estimation technique was employed to accurately identify five significant points of interest: the cricket's head and the proximal and distal ends of both the left and right antennae. This task was accomplished by utilizing

SLEAP (Pereira et al. (2022)), the state-of-the-art method for Animal Pose Estimation (APE), in order to streamline the labeling process and construct the neural network. To assess the model's ability to generalize, two datasets were created by iteratively labeling subsets of the videos to form training and validation sets. Subsequently, a grid search was carried out to optimize the model's parameters. Finally, the most effective model was utilized to predict the locations of the keypoints for each frame of all the videos, yielding a sequence of data indicating their precise locations.

2.3.1 Human-in-the-Loop Labeling

To generate the labels required for the SLEAP (Pereira et al. (2022)) training process, we executed the methodology proposed by Pereira et al. (2018), which highlighted that using a human-in-the-loop strategy for label creation along with network predictions significantly reduces human-only labeling time. The process is illustrated in Figure 3 and includes an initial labeling phase where we labeled 210 frames from 42 distinct videos (14 for each stimulus, with 5 frames per video) for the training set and 60 frames from 12 other videos (4 for each stimulus, again with 5 frames per video). We subsequently trained a model with a UNet backbone (Ronneberger et al. (2015)) using these sets by applying the parameters specified in Table 1, and implementing data augmentation by rotating the image between -180 and 180 degrees, as recommended by Pereira et al. (2022) for top viewpoint recordings. We utilized the model generated to predict 20 new frames for each video to augment both the training and validation sets. Before creating the datasets to be utilized for the following network training, we corrected the predicted frames.

In order to compare each iteration of labeling and training, we fixed the validation set at 300 frames after the initial iteration and never increased it again. This allowed us to evaluate the validation set for different iterations and determine if additional labeling was necessary to enhance our model. A total of 8 iterations were carried out, which resulted in a total of 5460 frames used for the training set before we observed no further increase in the mean Average Precision (mAP) relative to the validation set.

2.3.2 Grid Search for Parameter Optimization

After obtaining the training test consisting of 5460 frames, corresponding to 390 frames for 42 videos, a grid search was executed to determine the optimal configuration of hyperparameters that would enable more precise keypoints detection. To streamline this process and reduce the time taken, only the hyperparameters that had the greatest potential to impact the network were chosen: input scaling, max stride, and the number of filters. Input scaling was tested with values of 0.7, 0.8, 0.9, and 1.0, as an increase in this value can decrease the likelihood of finer features, such as the distal ends of cricket antennae that are characterized by 4 ± 1 pixels in width, being removed by downsampling the image. The max stride was tested with values of 32 and 64, where a larger value resulted in a larger receptive field but also increased the number of trained parameters in the network. The number of filters was tested with values of 32 and 64, which signifies the initial number of filters present in the first block of the UNet encoder (Ronneberger et al. (2015)). A higher number of filters can enhance the representational abilities of the network, but it can also increase memory usage and

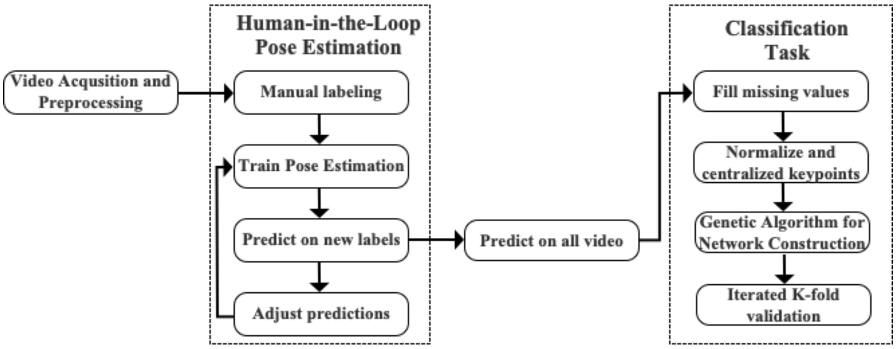

Video Acqusition and Preprocessing
Human-in-the-Loop Pose Estimation
Manual labeling
Train Pose Estimation
Predict on new labels
Adjust predictions
Predict on all video
Classification Task
Fill missing values
Normalize and centralized keypoints
Genetic Algorithm for Network Construction
Iterated K-fold validation

Fig. 3: Illustration of the complete workflow followed in this article. The workflow can be broadly divided into two parts, with the first part defined as Human-in-the-Loop Pose Estimation, wherein a model is trained to estimate the position of the antennae. The second part involves the use of a genetic algorithm to determine the optimal model architecture for classifying the behavioral interactions, which is subsequently evaluated using Iterated K-fold validation.

Table 1: Hyperparameters used for the network during the Human-in-the-Loop pose estimation phase.

Hyperparameter	Value
Max stride	64
Filters	64
Filters rate	2
Middle block	True
Up interpolate	True
Sigma	2.5
Output stride	2
Input scaling	0.7
Batch size	8
Epochs	400
Plateau min. delta	1e-08
Plateau patience	20

runtime. The training and subsequent operations described in this article were carried out using an NVIDIA A100 GPU with 40GB of GPU memory housed within a DGX A100.

2.3.3 Performance Metric for Pose Estimation

To evaluate network performance, we utilized the mean Average Precision (mAP) metric as described in SLEAP (Pereira et al. (2022)). This metric is based on the Object Keypoint Similarity (OKS) scores introduced by Ronchi and Perona (2017), which measure the similarity between ground truth (GT) and predicted object keypoints. The mAP metric involves classifying each pair of GT and predicted instance as a true positive (TP) or false positive (FP) based on the OKS score, using predetermined thresholds of 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95. For each threshold, precision is calculated as TP/(TP + FP), with predictions sorted by their OKS and cumulative TPs and FPs determined for each prediction. From these partial TPs and FPs, recall and precision values are derived for each pair of GT and predicted instance. Subsequently, a set of 101 recall thresholds with even spacing from 0 to 1 is defined, and the best precision value for samples below each recall threshold is obtained, resulting in 101 precision values. The average precision is then computed as the mean of all 101 precision values. This process is repeated for all ten OKS thresholds, and the final mAP is obtained as the average of the average precision overall thresholds.

The OKS was coded in its standard form:

$$OKS(X,\hat{X}) = \sum_{i=1}^{N} \exp\Big(-\frac{\|X_i - \hat{X}_i\|_2^2}{2\alpha\sigma_i^2}\Big)\delta_i\big(\sum_{i=1}^{N}\delta_i\big)^{-1} \tag{1}$$

In the calculation of OKS, the ground truth and predicted instance coordinates are denoted as X and $\hat{X}$, respectively, for an instance with N nodes. The visibility of each node is indicated by δ_i, with a value of 0 if the node is missing from the ground truth instance. The inner term of the OKS score expresses the distance between the predicted and ground truth coordinates as the posterior of a Gaussian distribution with two scaling terms: α and σ_i. Specifically, α denotes the bounding box area occupied by the GT instance, while σ_i is the uncertainty factor, which is set to 0.025 for all measurements, equivalent to the uncertainty in labeling human eyes (Pereira et al. (2022)).

2.4 Stimuli Classification Network

Once the pose estimation model had been trained, the next step was to classify the obtained sequences to predict the type of interaction (i.e., control, sucrose, ammonia). To achieve this task, we utilized neural network techniques related to processing sequences. Specifically, we developed two types of architectures: one based on one-dimensional convolution (Kim (2014)), and the other based on Recurrent Neural Networks (RNNs) that employed Long Short-Term Memory (LSTM) layers (Hochreiter and Schmidhuber (1997)) and Gated Recurrent Unit (GRU) layers (Cho et al. (2014)), in addition to bidirectional layers (Schuster and Paliwal (1997). In this section, we describe how the sequences were preprocessed and shaped, the strategy employed to autonomously create the networks using a genetic algorithm, and how the performance of the obtained networks was evaluated.

2.4.1 Sequence Preprocessing

After pose estimation, each video was converted into sequences that can be characterized by a shape (keypoints, frames), where the number of keypoints is fixed at 10, denoting the positions on the x- and y-axes of the head, right and left proximal and distal ends of the antennae. The number of frames in a sequence is equal to the interaction period, which is set to 3480. Due to the presence of NaN values in the sequences, a filling strategy was applied to handle missed values for a certain keypoint at a particular position t:

Due to the presence of NaN values in the sequences, a filling strategy was applied to handle missed values. For a certain missing keypoint at a frame t, the following equation defines how it is filled:

$$v_t = \frac{\alpha v_{t+k} + v_{t-1}}{1+\alpha}, \qquad \alpha = 1/k \tag{2}$$

In Equation 2, k identifies the first subsequent frame from the frame t with a non-NaN value for the keypoint under consideration. We also addressed the case where the value for $t = 0$ is NaN by setting it to the first subsequent non-NaN value for that specific keypoint.

The sequences were then centered on the head by moving it to the center of the Cartesian plane and translating the other points appropriately. This operation allows for the focus on the movement of the antennae, breaking the relation with the position of the cricket inside the Petri dish. After this transformation, the head-related values within the sequences were set to zero and subsequently removed, resulting in sequences with a shape of (8, 3480).

2.4.2 Genetic Algorithm for Neural Network Construction

In a study conducted by Cordeiro et al. (2021), it was demonstrated that the utilization of genetic algorithms to search for optimal architectures in one-convolutional models resulted in the development of highly accurate prediction models at a relatively lower cost when compared to other approaches, such as greedy, Bayesian, hyperband or random for network search.

This genetic approach can be described in the following recursive sequence (Abo-Hammour et al. (2013)):

- *Initialization.* An initial population was randomly generated with a size of 250 individuals, i.e., chromosomes, to trade off execution time and convergence (Abu Arqub et al. (2012)). We distinguished in two types of chromosomes based on our objective, either constructing one-convolutional or recurrent neural network. The chromosomes that comprise the one-convolutional network are constructed with a series of 56 real-coded genes divided in two blocks. The first block consists of six genes repeated five times to indicate: (1) whether the convolutional block is present (0 if absent, 1 if present); (2) the number of filters for the one-convolutional layer (ranging from 16 to 1024); (3) the presence of the batch normalization layer (0 if absent, 1 if present); (4) the activation function to be used (0: sigmoid, 1: swish, 2: tanh, 3: relu, 4: gelu, 5: elu, 6: leaky relu); (5) the presence of dropout (0 if absent, 1 if present); and (6) the dropout rate (ranging from 0 to 0.5, with consideration only given to multiples of 0.05). Subsequently, a gene is used to indicate the type of connection between the convolutional and fully connected layers, with a value of 0 indicating `Flatten` and 1 indicating `GlobalAveragePooling1D`. The second block, also repeated five times, comprises five genes that indicate: (1) whether the fully connected block is present (0 if absent, 1 if present); (2) the number of units (ranging from 3 to 512); (3) the activation function to be used; (4) the presence of dropout; and (5) the dropout rate. The chromosomes that comprise the RNN, on the other hand, consist of 50 real-coded genes. On the other hand, the RNN chromosomes consist of 50 real-coding genes with a slightly different configuration. The first block, also repeated five times, includes five genes that signify the presence of the RNN block, the use of a bidirectional layer, the type of RNN (LSTM or GRU), the number of units (16 to 1024), and the activation function. Unlike the one-convolutional network, there is no need for a gene related to the connection between convolutional and fully connected layers in the RNN.

- *Evaluation.* In a genetic algorithm, the evaluation is performed through a objected function called *fitness function*. In our experiment, we proposed the following fitness function that requires maximization:

$$\text{fit(gene)} = \begin{cases} -10 \cdot (1 - \text{train\_accuracy}) & \text{if } a \\ -15 & \text{if } b \\ -20 & \text{if } c \\ -\text{val\_loss} & O/W \end{cases} \tag{3}$$

  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 devise a fitness function, as opposed to solely minimizing the validation loss, stems from the recognition that in experiments with limited data and inherent complexity, such as ours, it is possible for a model to achieve a validation loss that is similar to, or even lower than, models with better validation accuracy and exceeding that of random guessing. To address this challenge, the fitness function was constructed to consider the training accuracy, providing an additional metric for assessing the network's quality. Higher training accuracy values yield fitness values closer to those derived from the validation loss, indicating a type of network quality that can be utilized in subsequent genetic algorithm iterations. Moreover, the `train_accuracy<val_accuracy` check is incorporated to prevent the genetic algorithm from overfitting on the validation accuracy, which may impede the ability to generalize effectively and harm the training. Lastly, we verify if the training accuracy value is below 0.1 and set a default value in such cases. This measure was taken to avoid any false indications of good models using the first case in Equation 3.
- *Selection.* The selection algorithm implemented in the genetic algorithm is tournament selection (Rudnick et al. (1997)), which operates by randomly selecting a fixed number of individuals, in our case two, from the population and subsequently choosing the most fit individual from this group to add to the mating pool. Moreover, in addition to tournament selection, the genetic algorithm also employs elitism as a selection strategy. This strategy involves preserving the top 10 individuals from the current population in the succeeding generation. The use of elitism ensures that the most exceptional individuals are given the opportunity to pass on their favorable

traits to future generations, which enhances the chances of achieving the desired solution.

- *Crossover.* The genetic algorithm's crossover algorithm is bounded Simulated Binary Crossover (bSBX), a bounded variant of Simulated Binary Crossover (SBX) introduced by Deb and Agrawal (1995). The probability value of crossover is set to 0.9.
- *Mutation.* The function applied for mutation is bounded polynomial mutation with a probability value of 0.5, a mutation operator that utilizes a polynomial function for probability distribution and is bounded to restrict the extent of the changes in the chromosome's value.
- *Termination.* Each genetic algorithm applications ran for 50 epochs before terminating.

To enhance the efficiency of the genetic algorithm execution, a function was developed to verify whether a constructed model had been previously trained, thereby avoiding redundant computations. Each model was trained for 400 epochs with a batch size of 8, utilizing the Adam optimizer (Kingma and Ba (2014)) with an initial learning rate of 1e-4. SLEAP's default learning rate decay strategy was adopted, incorporating a patience of 20 epochs and a minimum delta of 1e-8. To counteract the risk of overfitting, early stopping was employed, terminating the training process when the validation loss failed to decrease for 50 epochs. In addition, data augmentation techniques were applied, including the addition of Gaussian noise.

2.4.3 Performance Metric for Classification

To assess the performance of the neural network classifiers constructed by the genetic algorithm, accuracy was chosen as the evaluation metric. As our dataset is balanced, accuracy, defined by Equation 4, provides a measure of the model's ability to correctly classify or predict the output based on a given set of inputs. Expressed as a percentage, higher values indicate better performance.

$$Accuracy = \frac{TP}{\text{Total number of predictions}} \tag{4}$$

Once the optimal models were identified for both the convolutional and RNN cases using the genetic algorithm, iterated K-fold validation was employed to evaluate their effectiveness. To account for the limited data available, the dataset was randomly shuffled and split into training and validation sets, with K-fold validation executed multiple times using a number of iterations set to 10 and k set to 4. This approach ensures an accurate assessment of the model's performance, with the final score calculated as the mean of the accuracies achieved across each K-fold validation run.

3 Results

3.1 Pose Estimation Results

To determine the optimal configuration of parameters for our experiment, we utilized the grid search approach outlined in subsubsection 2.3.2. Table 2 displays the results of the model testing process, which indicates that the model with a maximum stride of 64, a filter count of 64, and input scaling set to 1.0 achieved the highest validation mAP score of 0.837392.

Table 2: Grid search results for pose estimation.

Max Stride	Filters	Input scaling	Train mAP	Val mAP
32	32	0.7	0.837606	0.776263
32	32	0.8	0.825532	0.782753
32	32	0.9	0.825385	0.764668
32	32	1.0	0.845399	0.800341
32	64	0.7	0.864492	0.795866
32	64	0.8	0.865248	0.809098
32	64	0.9	0.863264	0.799638
32	64	1.0	0.885996	0.829130
64	32	0.7	0.824700	0.748723
64	32	0.8	0.843466	0.775747
64	32	0.9	0.777974	0.643496
64	32	1.0	0.849048	0.793634
64	64	0.7	0.867076	0.804768
64	64	0.8	0.805469	0.732222
64	64	0.9	0.863854	0.808308
64	64	1.0	0.892736	0.837392

3.2 Performance of the Generated Classifiers

The implementation of genetic algorithms for the development of convolutional and recurrent neural networks resulted in the generation of two distinct structures. The convolutional neural network (CNN) was composed of a convolutional layer featuring 821 filters, followed by a batch normalization layer and a hyperbolic tangent activation layer. This was then followed by another convolutional layer containing 821 filters and an exponential linear unit (ELU) activation layer. A dropout layer with a rate of 0.2 was then introduced, followed by a final convolutional layer comprising of 483 filters and a hyperbolic tangent activation layer. The output of the convolutional layers was flattened, and no fully connected layers were present except for the softmax classifier for the output. On the other hand, the recurrent neural network (RNN) was structured as a bidirectional LSTM layer with 707 units and an ELU activation function, followed by a gated recurrent unit (GRU) layer with 660 units and a leaky rectified linear unit (ReLU) activation function. A bidirectional GRU layer with 469 units and a leaky ReLU activation function was then added, followed by a dense layer with 138 units and a Gaussian error linear unit (GELU) activation function. A dropout layer with a rate of 0.2 was then inserted, followed by a dense layer comprising of 150 units and a leaky ReLU activation function. The validation accuracy of the CNN and RNN models was 58.33% and 50%, respectively.

The outcomes of the identified models resulting from 10 iterations of the 4-fold validation are delineated in Table 3. These outcomes are juxtaposed with the baseline results obtained through the utilization of distinct one-convolutional ResNet (RN) architectures. This comparison serves to accentuate the enhancements achieved through the employment of the genetic algorithm. Our developed *cnn* and *rnn* models exhibit mean accuracy values of 45.33% ± 5.85% and 44% ± 6.6%, respectively. This indicates a marginal superiority of the generated CNN over the RNN. In contrast with the ResNet models, our models exhibit greater robustness across all iterations, consistently avoiding instances where the mean accuracy value equates to the random guess of 0.33, denoting chance-level accuracy with respect to the number of classes.

Model	**Iter 1**	**Iter 2**	**Iter 3**	**Iter 4**	**Iter 5**	**Iter 6**	**Iter 7**	**Iter 8**	**Iter 9**	**Iter 10**	**Avg**
cnn	0.58	0.40	0.37	**0.48**	**0.52**	0.45	**0.43**	**0.45**	**0.43**	0.42	**0.45**
rnn	**0.60**	0.38	0.36	0.36	0.47	**0.48**	**0.43**	**0.45**	0.40	**0.45**	0.44
RN18	0.46	**0.41**	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.35
RN34	0.38	0.33	0.33	0.33	0.41	0.33	0.38	0.33	0.33	0.33	0.35
RN50	0.33	0.33	0.36	0.33	0.33	0.33	0.33	0.33	0.45	0.36	0.36

Table 3: Mean accuracy values for each iteration of the 10 Iterated 4-fold validation, and the average of all the iteration. (RN stands for ResNet)

4 Discussion

In this study, Animal Pose Estimation (APE) and genetic algorithms were employed to predict the type of interaction between a cricket (*Acheta domesticus*) and three possible stimuli (i.e., nothing, sucrose or ammonia powders). The genetic algorithms were utilized with chromosomes representing the architecture of possible convolutional and recurrent neural networks. The pose estimation network was obtained using SLEAP, and it achieved high mean average precision (mAP) results in detecting the head, proximal, and distal antenna ends. However, some difficulties arose when detecting the distal ends, particularly when the crickets were positioned near the wall of the Petri dish, causing overlapping and reflection issues. Figure 4 depicts two frames where our model misplaced one of the distal ends alongside the corresponding correct keypoint location. The correct location was determined by analyzing the previous and subsequent frames to understand the movements of the antennae. Although SLEAP does not use temporal information, it could be valuable in improving detections in situations where overlapping and occlusions are frequent. To this aim, recent research in APE has been directed towards the development of neural networks leveraging that information (Russello et al. (2021)), but proposing simpler models with respect to the one here employed in order to limit the computational power required for training. As for our pose estimation model hyperparameters, their high values indicate the complexity of the task at hand. This is highlighted by the input scaling value, which indicates that no downscaling operation was employed to preserve all information from the pixels, underscoring the challenges associated with detecting subtle features such as the distal ends of the antennae.

On the contrary, the classification task yielded less satisfactory results. Unlike the literature surrounding human pose estimation and movement comprehension (Núñez

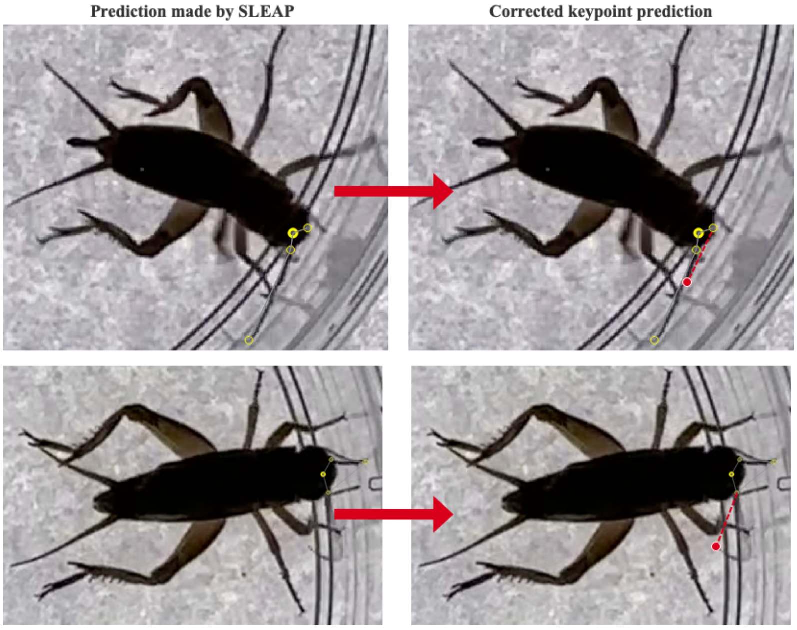


Fig. 4: Two examples in which the model has erroneously labeled the distal ends of the antennae. The top one showcases an occurrence in which the model has erroneously identified a frame by misplacing the right and left distal ends in a single location. The one below depicts the visibility of the right antenna's distal end, which, despite this fact, has still been incorrectly positioned by the network. This anomaly may be attributed to the resemblance in shapes and black markings of the other limbs.

et al. (2018), Carrara et al. (2019)), the one-convolutional neural network produced better average results in our study than the commonly used RNNs. The suboptimal performance of the classification task could be attributed to the animals' limited attention span and behavioral variations, which pose challenges in utilizing them as biosensors (Oh et al. (2015)). Despite this limitation, it is noteworthy that animal training could be a crucial component to incorporate prior to the pose estimation phase, particularly when accessing animals' innate desire to detect specific substances, such as in our case. Animal training is commonly conducted in biosensor development (Oh et al. (2015)), Romano et al. (2022), and Matsumoto (2022) provides examples of this approach using crickets, which could be tested in our possible future research.

Despite the results obtained, our proposed automated workflow for the development of Biohybrid Intelligent Sensing Systems (BISS) holds great potential for a variety of applications. Firstly, the proposed workflow can be applied to different scenarios, making it a versatile solution not limited to crickets. Secondly, it has the potential to address certain limitations associated with the use of animal biosensors. These limitations include the introduction of errors stemming from human observation and interpretation, which can be avoided through the automated workflow, thereby facilitating method standardization. Thirdly, the proposed workflow uses only recordings without the need for the installation of devices connected to the animal, making it a non-invasive and ethical alternative. Lastly, the use of lifeforms in the proposed workflow results in a longer maintenance break compared to other technological systems, providing a more efficient solution that requires less frequent maintenance (Rajewicz et al. (2022)).

5 Conclusion

In conclusion, the presented research is characterised by a dual-part workflow used to correlate crickets' antennal movement to specific stimuli (i.e., ammonia, A, and sugar, S, powders) , where the initial phase involves the utilisation of pose estimation techniques. This phase employs SLEAP, leveraging a U-Net backbone, to accurately determine the antennae's positions. The subsequent phase integrates genetic algorithms, facilitating the construction of unique one-convolutional and recurrent neural networks. These networks are adeptly tailored to address the intricate classification task at hand. Remarkably, the outcomes of this study are juxtaposed against ResNet (RS) one-convolutional models, showcasing the inherent potential of genetic algorithms in conquering intricate challenges. Through the employment of genetic algorithms, we've achieved models that are both more robust and precise compared to well-established architectures, underscoring the efficacy of this approach.

In the realm of animal biosensors, a burgeoning fascination spans across diverse domains, encompassing agriculture, medicine, and environmental surveillance. Within this context, our research presents a pivotal endeavour towards shaping a workflow that bridges animal behaviour with chemical cues, culminating in the establishment of Biohybrid Intelligent Sensing Systems (BISS). The practical manifestation of our workflow unfolds through the exploration of cricket antennae movement, responding to specific stimuli—ammonia and sugar powders. This exploitation of their inherent olfactory prowess underscores the viability of this approach. Notably, our study pioneers the release of a groundbreaking dataset, housing cricket recordings tailored for pose estimation analyses. This endeavour serves as a cornerstone, charting the course for forthcoming research ventures in this domain. The implications of our findings could transcend their immediate scope, resonating broadly across the landscape. The path we tread paves the way for the evolution of animal biosensors and bioindicators, rooted in deep learning paradigms and proficient in extracting insights exclusively from recordings. This methodology holds the promise of nurturing ethical and ecologically sound research practices, circumventing invasive methodologies and toxic substances.

Acknowledgments

This work was partially carried out in the framework of the H2020 FETOPEN Project "Robocoenosis—ROBOts in cooperation with a bioCOENOSIS" [899520]. The founder had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Declarations

Conflict of interest. The Authors listed in this article declare that they have no conflict of interest.

Ethical approval. The present investigation was conducted in full compliance with the directives set forth by the guidelines for the treatment of animals in behavioral research and teaching, as prescribed by the Association for the Study of Animal Behaviour (ASAB) and the Animal Behavior Society (ABS) in the year 2014. The Italian regulations (No. 116192, 1927) and the laws enacted by the European Union (European Commission's directive of 2007) were also strictly adhered to. All experimental procedures comprised behavioral observations exclusively, and no specific authorizations were deemed necessary in the jurisdiction where the study was conducted.

Availability of data and materials. The datasets containing all the recordings used and the obtained pose estimation predictions, along with the post-processed predictions, can be downloaded from Google Drive.

Code availability. The code used in this article is publicly available on GitHub together with demos and a instruction on how to use the code.

References

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Laak, J.A.W.M., Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. CoRR **abs/1702.05747** (2017) 1702.05747

Jha, K., Doshi, A., Patel, P., Shah, M.: A comprehensive review on automation in agriculture using artificial intelligence. Artificial Intelligence in Agriculture **2**, 1–12 (2019) https://doi.org/10.1016/j.aiia.2019.05.004

Couzin, I.D., Heins, C.: Emerging technologies for behavioral research in changing environments. Trends in Ecology & Evolution **38**(4), 346–354 (2023) https://doi.org/10.1016/j.tree.2022.11.008 . Special issue: Animal behaviour in a changing world

Khanzode, K.C.A., Sarode, R.D.: Advantages and disadvantages of artificial intelligence and machine learning: A literature review. International Journal of Library & Information Science (IJLIS) **9**(1), 3 (2020)

Rolnick, D., Donti, P.L., Kaack, L.H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A.S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A., Maharaj, T., Sherwin, E.D., Mukkavilli, S.K., Körding, K.P., Gomes, C.P., Ng, A.Y., Hassabis, D., Platt, J.C., Creutzig, F., Chayes, J.T., Bengio, Y.: Tackling climate change with machine learning. CoRR **abs/1906.05433** (2019) 1906.05433

Romano, D., Donati, E., Benelli, G., Stefanini, C.: A review on animal–robot interaction: from bio-hybrid organisms to mixed societies. Biological cybernetics **113**, 201–225 (2019)

Oh, Y., Lee, Y., Heath, J., Kim, M.: Applications of animal biosensors: A review. IEEE Sensors Journal **15**, 637–645 (2015)

Pickel, D., Manucy, G.P., Walker, D.B., Hall, S.B., Walker, J.C.: Evidence for canine olfactory detection of melanoma. Applied Animal Behaviour Science **89**(1), 107–116 (2004) https://doi.org/10.1016/j.applanim.2004.04.008

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

Olson, D., Rains, G.: Use of a parasitic wasp as a biosensor. Biosensors **4**(2), 150–160 (2014)

Saha, D., Mehta, D., Atlan, E., Chandak, R., Traner, M., Lo, R., Gupta, P., Singamaneni, S., Chakrabartty, S., Raman, B.: Explosive sensing with insect-based biorobots. bioRxiv (2020)

Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., Bethge, M.: Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Nature neuroscience **21**(9), 1281–1289 (2018)

Pereira, T.D., Tabris, N., Matsliah, A., Turner, D.M., Li, J., Ravindranath, S., Papadoyannis, E.S., Normand, E., Deutsch, D.S., Wang, Z.Y., *et al.*: Sleap: A deep learning system for multi-animal pose tracking. Nature methods **19**(4), 486–495 (2022)

Luxem, K., Mocellin, P., Fuhrmann, F., Kürsch, J., Miller, S.R., Palop, J.J., Remy, S., Bauer, P.: Identifying behavioral structure from deep variational embeddings of animal motion. Communications Biology **5**(1), 1267 (2022)

Fang, C., Zhang, T., Zheng, H., Huang, J., Cuan, K.: Pose estimation and behavior classification of broiler chickens based on deep neural networks. Computers and Electronics in Agriculture **180**, 105863 (2021)

Loudon, C., Bustamante, J., Kellogg, D.W.: Cricket antennae shorten when bending (acheta domesticus l.). Frontiers in Physiology **5** (2014)

Draft, R.W., McGill, M., Kapoor, V., Murthy, V.N.: Carpenter ants use diverse antennae sampling strategies to track odor trails. Journal of Experimental Biology **221** (2018)

Yamawaki, Y., Ishibashi, W.: Antennal pointing at a looming object in the cricket acheta domesticus. Journal of insect physiology **60**, 80–91 (2014)

Pereira, T.D., Aldarondo, D.E., Willmore, L., Kislin, M., Wang, S.S.-H., Murthy, M., Shaevitz, J.W.: Fast animal pose estimation using deep neural networks. Nature

Methods **16**, 117–125 (2018)

Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. ArXiv **abs/1505.04597** (2015)

Ronchi, M.R., Perona, P.: Benchmarking and error diagnosis in multi-instance pose estimation. 2017 IEEE International Conference on Computer Vision (ICCV), 369–378 (2017)

Kim, Y.: Convolutional neural networks for sentence classification. In: Conference on Empirical Methods in Natural Language Processing (2014)

Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997)

Cho, K., Merrienboer, B., Gülçehre, Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Conference on Empirical Methods in Natural Language Processing (2014)

Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**, 2673–2681 (1997)

Cordeiro, J.R., Raimundo, A., Postolache, O.A., Sebastião, P.J.A.: Neural architecture search for 1d cnns—different approaches tests and measurements. Sensors (Basel, Switzerland) **21** (2021)

Abo-Hammour, Z., Alsmadi, O., Momani, S., Abu Arqub, O., et al.: A genetic algorithm approach for prediction of linear dynamical systems. Mathematical Problems in Engineering **2013** (2013)

Abu Arqub, O., Abo-Hammour, Z., Momani, S., Shawagfeh, N., *et al.*: Solving singular two-point boundary value problems using continuous genetic algorithm. In: Abstract and Applied Analysis, vol. 2012 (2012). Hindawi

Rudnick, E.M., Patel, J.H., Greenstein, G.S., Niermann, T.M.: A genetic algorithm framework for test generation. IEEE Transactions on computer-aided design of integrated circuits and systems **16**(9), 1034–1044 (1997)

Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9** (1995)

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)

Russello, H., Tol, R., Kootstra, G.: T-leap: occlusion-robust pose estimation of walking cows using temporal information. Comput. Electron. Agric. **192**, 106559 (2021)

Núñez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Vélez, J.F.: Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit. **76**, 80–94 (2018)

Carrara, F., Elias, P., Sedmidubsky, J., Zezula, P.: Lstm-based real-time action detection and prediction in human motion streams. Multimedia Tools and Applications **78**, 27309–27331 (2019)

Romano, D., Rossetti, G., Stefanini, C.: Learning on a chip: Towards the development of trainable biohybrid sensors by investigating cognitive processes in non-marine ostracoda via a miniaturised analytical system. Biosystems Engineering **213**, 162–174 (2022) https://doi.org/10.1016/j.biosystemseng.2021.11.004

Matsumoto, Y.: Learning and memory in the cricket gryllus bimaculatus. Physiological Entomology **47**, 147–161 (2022)

Rajewicz, W., Romano, D., Varughese, J.C., Schmickl, T., Thenius, R.: Lifeforms potentially useful for automated underwater monitoring systems. The 2022 Conference on Artificial Life (2022)