KI - Künstliche Intelligenz
https://doi.org/10.1007/s13218-023-00812-x

TECHNICAL CONTRIBUTION

Towards noise robust acoustic insect detection: from the lab to the greenhouse

Jelto Branding[1] · Dieter von Hörsten[1] · Jens Karl Wegener[1] · Elias Böckmann[2] · Eberhard Hartung[3]

Received: 5 December 2022 / Accepted: 27 September 2023

Abstract
Successful and efficient pest management is key to sustainable horticultural food production. While greenhouses already allow digital monitoring and control of their climate parameters, a lack of digital pest sensors hinders the advent of digital pest management systems. To close the control loop, digital systems need to be enabled to directly assess the state of different insect populations in a greenhouse. The presented article investigates the feasibility of acoustic sensors for insect detection in greenhouses. The study is based on an extensive dataset of acoustic insect recordings made with an array of high-quality microphones under noise-shielded conditions. By mixing these noise-free laboratory recordings with environmental sounds recorded with the same equipment in a greenhouse, different signal-to-noise ratios (SNR) are simulated. To explore the possibilities of this unique and novel dataset, two deep-learning models are trained on this simulation data. A simple spectrogram-based model represents the baseline for a comparison with a model capable of processing multi-channel raw audio data. Making use of the unique possibility of the dataset, the models are pre-trained on clean data and fine-tuned on noisy data. Under lab conditions, results show that both models can make use of not just insect flight sounds but also the much quieter sounds of insect movements. First attempts under simulated real-world conditions showed the challenging nature of this task and the potential of spatial filtering. The analysis enabled by the proposed methods for training and evaluation provided valuable insights that should be considered for future work.

Keywords Insects · Audio · Acoustic · Identification · Recognition · Classification · Low-noise microphone · Microphone array · Anechoic box · Pest detection · Horticulture · Deep learning · WaveNet · Spectrogram · Raw audio · Neural beamforming · Background noise · Noise simulation

1 Introduction

Acoustic insect recognition could be a key part of developing a digital insect sensor unit [14]. Currently, the process of evaluating an insect population, for example in a greenhouse, is a labour-intensive task that requires expert knowledge. Digital insect sensor units would permit digital systems to directly assess the state of different insect populations in a crop stand. Such sensors would enable scientists and plant growers to greatly improve the quality and temporal resolution of insect population data. Instead of counting insects manually once a week or so, digital sensors would allow for continuous monitoring. Better and bigger population datasets of higher temporal resolution would enable the development of better population prediction models. Over the course of a growing season, a dataset would even allow the quantitative evaluation of different pest management measures and ultimately allow the development of sophisticated pest management aid systems. The sooner a critical pest population in a greenhouse can be identified and the sooner the right management measures can be applied, the smaller those measures need to be. While small infestations can often be treated with the release of beneficial insects, bigger pest

✉ Jelto Branding
jelto.branding@julius-kuehn.de

[1] Institute for Application Techniques in Plant Protection, Julius Kühn Institute (JKI), Messeweg 11/12, 38104 Braunschweig, Germany

[2] Institute for Plant Protection in Horticulture and Urban Green, Julius Kühn Institute (JKI), Messeweg 11/12, 38104 Braunschweig, Germany

[3] Institute of Agricultural Process Engineering, Christian-A lbrechts-Universität zu Kiel, Max-Eyth-Str. 6, 24118 Kiel, Germany

Published online: 30 October 2023

populations typically require the use of chemical pest management measures. Therefore, digital aid systems could play an important role in making the pest management of the future more efficient and, ultimately, allow for more sustainable food production.

While within the broader research field of insect sounds different specific groups of insects have been studied in depth [12], examples of signal classification are rare [18]. The classification of low-level insect sounds in a greenhouse is challenging mostly because of the low signal-to-noise ratio (SNR). This is also true for human speech recognition in far-field conditions, where the speaker is further away from the microphone. Highly sophisticated products, like Amazon Echo Dot or Google Home [10], deal with these challenging conditions by not only using one microphone, but instead using an array of microphones. The use of multiple microphones allows these systems to apply spatial filtering techniques to enhance the target signal quality. In classical signal processing, this spatial filtering is performed by so-called beamforming algorithms. Latest advances in this field saw the use of so-called neural beamforming (NBF) systems jointly trained with complex speech recognition models [5, 23].

A key difference between human speech and insect sound signals is the necessary temporal context for classification. While the correct transcription or translation of a human spoken word may rely on the understanding of an entire paragraph, the simpler nature of insect sound allows for the use of much shorter signals. This difference in necessary temporal context also explains the different model types dominantly used for human speech and bird song-related tasks. While human speech tasks are often solved by variants of long-short-term-memory models (LSTM), bird songs can be successfully recognized by much simpler convolutional neural network (CNN) models.

Another difference separating insect sounds from both human speech and bird song is the frequency range. The sounds that need to be used to identify insects are predominantly flight sounds. As the frequency of flight sounds is upward restricted by the max speed of the muscle contractions moving the wing, found to be 1000 Hz [25], these flight sounds typically have a base frequency way below bird song or human speech. While bird songs typically range from 3 kHz to 5 kHz and human speech is transmitted adequately via telephone using a frequency range of 300 Hz to 3400 Hz [29], the flight sound base frequency of insects typically found in a European greenhouse ranges between roughly 60 Hz and 200 Hz. When linking this information with the information that most of the environmental noise is concentrated in the lower frequency range [2], often described as 1/f-noise, one can see the challenging nature of the task of acoustic insect recognition in the presence of environmental sounds.

While the recognition of insects from recordings made in an acoustic laboratory might be of scientific interest, only recognition despite the presence of environmental noises is of interest for practical applications. Based on the latest findings in the field of human speech recognition, the hypothesis is that the use of multiple microphones should benefit the task at hand.

Research goal

This leads to the question: How can microphones be used to identify insects? To answer this question, two main tasks have to be solved. First, a way to record different insect sounds has to be found. Second, an analysing tool has to be developed, that can use this sound data to differentiate the insects. The first task was solved by employing highest quality measurement microphones and recording in a noise-shielded environment. While this procedure is described in the following, the focus of this paper lies on the second task.

This paper reports on an effort to not only show the feasibility of recognizing insects by their sound, but to develop a system that can do so with significant environmental noise present in the recordings. The models presented try to combine the latest progress in different related fields to a unique solution special to this task. First, a baseline solution, consisting of a CNN model trained on the spectrograms of a single microphone, is presented. To improve upon this baseline solution, a second model, processing raw audio data from multiple microphones is introduced. The proposed architecture for this model consists of a NBF layer and a WaveNet classifier.

As this study relies on a unique and novel dataset of insect sounds and noise recordings, the next sections disclose the methods used to create this data. Considering the scope and focus of this paper, the method development process behind the method presented shall not be elaborated here, and instead will be published in detail in a separate publication to come.

2 Material and Methods

The process from recording insect sounds to their classification by a deep learning model involves many steps along the way. Following, the decisions made along this path shall be described beginning with the experimental setup used to record the different sounds and continuing through signal processing, towards the choice of deep learning models and the setup of their training pipelines.

2.1 Acoustic Recordings

Working with insect sounds presents not only a challenging classification task, but also a challenging acoustic recording

task. Next, the special equipment and experimental setup used to enable the reproducible generation of high-quality sound data are described.

2.1.1 Recording Hardware Setup

All recordings used in this study, either in the acoustic laboratory or the greenhouse, were made using an array of four low-noise measurement microphones. The microphones used were Brüel & Kjaer type 4955, each with a sensitivity of more than 1300 $mVPa^{-1}$ and a self-noise level of 6.5 dB(A). The microphones were assembled to form an array using a self-designed 3D-printed fixture made from PA-12. The fixture places the microphones in a star-like shape, with one microphone in the centre and the three others within a constant radius of 55 mm (centre to centre) around the middle one (see Online Resource 1).

The microphones were powered by a Brüel & Kjaer Nexus 2690 Conditioning Amplifier. This unit also offers an analogue low and high pass filter option, as well as an integrated analogue amplifier. The analogue high pass filter on the Nexus was set to 20 Hz and the low pass filter was set to 10 kHz. To minimize the noise introduced by the amplifier inside the Nexus, one has to employ a little trick. From the data sheet supplied with the Nexus, it is obvious that the self-noise introduced by the amplifier is minimised at gain levels that are a whole-number multiple of 10 dB. As the Nexus by default is set up to equalize the sensitivity of every channel by adjusting every channel's gain to pull all microphones onto the same output sensitivity, one has to bypass this function to be able to explicitly set the gain level for every channel. This can be done by setting the correction factor on the Nexus for each microphone to the inverse of its microphone's sensitivity. By setting the Nexus up this way and choosing an output level of 10 VPa^{-1} on the Nexus, the signal of every channel was amplified by 20 dB, while introducing the minimal possible amount of amplifier noise into the signal.

The analogue low and high pass filtered and amplified signal was then fed to an A-D-converter made by Roga Instruments, called DAQ4. The gain function on the DAQ4 was disabled, as the preliminary test showed that the amplification using the DAQ4 would result in more noise than using the same amplification level on the Nexus. Furthermore, the DAQ4 was set to AC-coupled mode and a digital high pass filter with 0.3 Hz was activated. All recordings were digitalized using a sample rate of 48 kHz. The DAQ4 was set up and recorded using the software DasyLab 2022 installed on a standard office notebook. The recordings were saved as single-precision (32-bit, floating-point) tdms-files, each 14:13 min long. The tdms-file-format is a file format for measurement data introduced by National Instruments, who offer a python library that allows for easy further processing and file-format conversion of these files.

2.1.2 Acoustic Lab Environment: Design of an Anechoic Box

To shield the recording environment from environmental noises, an anechoic box was built (pictured in Online Resource 2). Medium-density fibreboard (MDF) wood panels were used to build a double-wall structure made of two boxes. The inside box was made from 25 mm panels, while the outside box was made from 28 mm panels. The inside dimensions of the inner wooden box were chosen as 1170 mm × 960 mm × 750 mm.

To increase the noise shielding effect of the box and minimize reverberations inside the box, different foam layers were used. A 100 mm thick layer of high-density (120 $kg\,m^{-3}$) composite foam was used to absorb reverberating sound between the walls of the two boxes and increase the mass of the outside box walls to help absorb low-frequency environmental noises. The dimensions of the outside box were chosen so as to leave an air gap of 100 mm from the high-density foam to the inside box. The inner box was lined with three 50 mm thick layers of open-cell acoustic foam (MicroPor) to absorb reverberations and shield against high-frequency environmental noise. The different foam panels were attached to the wood panels using spray glue. The inner box was then suspended by springs from the top of the outside box, following the design presented by [15], in an effort to decouple the recording environment from building vibrations. The springs used were rated at a spring rate of 4.634 $N\,mm^{-1}$.

Special attention was given to the design of the box doors. While the inside box had a normally designed door, the outside box was split into the door and box in such a way that the remaining parts were narrow enough to fit through building doors. To minimize the acoustic leakage of the doors, the inside box door was built in a labyrinth-sealing manner, resulting in a three-layer pyramid of foam on the door panel. The outside box door was sealed by letting the foam overlap 120 mm from the door into the box body, also resulting in a labyrinth-type sealing.

The box was placed on rollers to facilitate movement and further improve isolation from building vibrations. The main part of the outside box was placed on four heavy wait transport rolls made from rubber. The outside box door was also placed on two of these rolls, supporting the door when opened. Cables were routed through the top of the boxes via holes cut into the two boxes. To reseal the holes, they were stuffed with foam after routing the cables through. The assembled box has a total mass of 620 kg not including the rolls, springs and bolts used for assembly. Preliminary experiments aiming to characterize the noise absorption performance of the box showed good isolation at frequencies above 100 Hz.

Experimental setup inside the box

Inside the box, an insect-rearing cage was placed directly in front of the microphone array. The cages used had a size of 32.5 cm × 32.5 cm × 77 cm and were made from all mesh material. Inside the rearing cage, a custom square plant pot made from PVC plates was made to exactly fit the rearing cage measurements. Within each plant pot, two small tomato plants were planted in diagonal corners. Having plants and soil inside the cage should allow the recording of sounds that stem from the interaction of insects with plants and reduce the amount of sounds that stem from insects interacting with the cage e.g. by walking on the mesh. After problems with fungus gnat infections in the potting soil, the soil used to plant the tomato plants was disinfected by microwaving it for 200 s at 900 W. To prevent dammed-up water, clay pellets were used at the bottom of the plant pot. In total six identical cages were set up as described and were inoculated with different pest- and beneficial insects.

The microphone array was mounted to a fixture for easy positioning. Using the fixture, the microphone array was placed about 30 cm above the box and insect cage floor. The microphones were placed directly facing the cage, very close to, but not touching, the cage mesh.

The light inside the box was provided by two types of LED stripes, to stimulate natural daytime behaviour in insects. The stripes were glued onto a small aluminium plate as a heat sink and placed on top of the cage. The LED stripes used produced visible light ($CRI = 95\,\%$) and UV light.

2.1.3 Insects Recordings

The choice of insects was founded on two thoughts. The ultimate purpose of this study is to build towards a system applicable in the context of horticulture. Therefore, first, the insects chosen for this study should likely be found in a European greenhouse. Second, the goal was to portray a range of different sound levels. As the investigated models are expected to show decreased performance, or even fail, in the classification of quiet insects in the presence of louder environmental sounds, selecting insects of different sound levels should allow for a more detailed analysis of model performances. For this study, the following five insects were selected from the insect recordings made in the anechoic box:

- *Bombus terrestris*
- *Palomena prasina*
- *Episyrphus balteatus*
- *Coccinella septempunctata*
- *Aphidoletes aphidimyza*

Of the selected insects, four are used in horticulture for pest management or pollination and only *P. prasina* is considered a pest. The *P. prasina* specimens used for the recordings were captured in nature as adults and were then fed and kept in a rearing cage. As the other insects are deliberately released in greenhouses for their benefits, they are commercially reared and were simply purchased from a local beneficial insect supplier.

The order in which the insects were listed above represents a subjective ranking of their sound level from high to low. *B. terrestris*, commonly known as a bumblebee, represents the top of the scale. These insects produce a loud constant hum during flight that is audible within a few meters range for a human ear. *P. prasina* is the biggest of the insects selected for this study, and therefore has a relatively loud flight sound. Compared to the bumblebee, however, it is already significantly quieter. Just like the flight of *P. prasina*, *E. balteatus* and *C. septempunctata* are audible to the human ear when in flight. Correlating to its size, *C. septempunctata* again seems a lot quieter than *E. balteatus*, to the point where it is only audible to the human ear in a very limited range. The quietest of the selected insects is *A. aphidimyza*. With its delicate wings, this small gall midge produces a high-pitched buzz, so quiet it is not audible to a human.

As the recording equipment used allows the recording of sounds even below the human threshold of hearing, the flight sounds of *A. aphidimyza* are clearly audible in the recordings. Furthermore, through the use of this highly specialised equipment, the sounds recorded are not limited to the flight sound most notable to humans. Listening to the recordings, there appear to be many sounds that must stem from other insect movements than flight. These sounds are a lot quieter and can be described as rattling or scratching noises.

Because of the high sensitivity of the microphones used and the considerable amount of gain applied to the signals, the sound quality would seem rough to someone used to listening to modern music recordings. Even though the equipment used allows to capture remarkably quiet sounds, this setup introduces a considerable amount of amplifier noise into the recordings. Figure 1 displays an example of a flight sound recording of each insect, picked from the dataset and depicted as a spectrogram.

By placing only insects of one species inside the anechoic box at a time, all sounds recorded in one session should be from this one insect species. Because it is impossible, even for expert entomologists, to assign all the sounds recorded to the correct insect species, a different approach to labelling is necessary. The presented experimental set-up, which physically ensures all sounds recorded in one session must come from one species, represents the only way to generate labelled insect sound recordings. This is especially true considering that the used recording equipment allows for the recording of sounds that are inaudible and therefore unknown to human ears.

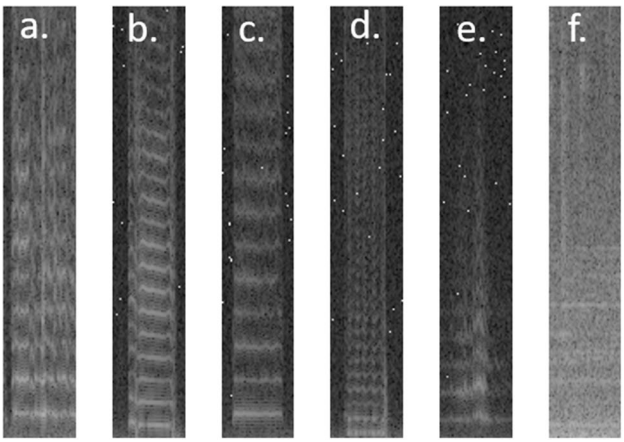


Fig. 1 Spectrogram of 2.5 s sound samples from the datasets, showing the frequency range of 50 Hz to 2000 Hz on the vertical axis over time on the horizontal axis. (a.) to (e.) are flight sound of *Bombus terrestris*, *Palomena prasina*, *Episyrphus balteatus*, *Coccinella septempunctata* and *Aphidoletes aphidimyza* respectively. (f.) is one of the background noise samples recoded in the greenhouse. These spectrograms where generated using the parameters and process described in Sect. 2.2.1

To minimize the influence of noise from people working in the building, the recordings were conducted overnight. The recording period started in the early afternoon and ended in the morning. The light inside the box was controlled via a time switch and set to turn off at 3:00 in the morning. To account for the night recordings, the light schedule in the climate chambers where the insect cages were kept when not recorded, was delayed by three hours. For every recording, only one cage containing 3 to 50 insects of one species was placed inside the anechoic box. Bigger, louder insects were recorded using only a few individuals, while smaller insects were recorded in higher numbers, to ensure that three to four nights of recording time would yield roughly the same amount of insect sound events in the recordings. In total, the insect sound data used in this study was recorded over 20 nights, amounting to 312 h of recordings.

2.1.4 Environmental Sound Recordings

To be able to simulate the sound conditions of insect sound recordings inside a greenhouse as well as possible, environmental sound recordings were made inside a real-world greenhouse. The greenhouse had a length of 18.55 m, a width of 9 m and a ridge height of 5 m and was constructed of metal with glass windows. As the greenhouse was made for cultivation in soil, the floor was covered with soil, except for four paths of rubber mats. Further, the structure contained metal constructions for fixing high-growing vegetables and an irrigation system. About a quarter of the area was planted with young pepper plants. Windows were opened and closed automatically by a climate control system. The greenhouse was located in the facilities of the JKI in Braunschweig and is used for research purposes.

The goal of these environmental sound recordings was to, if possible, not contain insect sounds. Therefore, the recordings were made at the time when the first plants were planted. Temperatures were still relatively low and so was the number of insects inside the greenhouse.

To build a diverse dataset, six different locations inside the greenhouse were recorded. At every recording location, two different orientations of the microphones were recorded, to vary the direction in which the constant noise sources would reach the microphones. The microphone array was mounted about one meter above the ground using a tripod. The measurement chain consisted of the B &K Nexus and DAQ4 and an office notebook, set up the same as for the laboratory recordings inside the anechoic box. For each of the 12 variants (6 locations $\times$ 2 orientations), a roughly 5 min long recording was recorded. While recording, the audible sound events were journalized by two people standing about 5 m from the microphones. The sound events that occurred during the recordings included:

- A constant, low-level humming sound produced by the climate chambers nearby
- Noise from the greenhouse windows opening and closing via motors
- Sound of workers working in the greenhouse next door
- Air and car traffic noise, including sirens and car horns
- Bird calls

Image (f.) in Fig. 1 shows a spectrogram illustrating a small section of one of the greenhouse recordings.

2.2 Models—Selection, Training and Evaluation Methods

Following, the two deep learning models used to explore the novel dataset are presented, before the intricacies of training on noisy data are discussed. This section is concluded with a description of the extra steps necessary for a meaningful evaluation of the investigated models under noisy conditions.

2.2.1 Model Selection

Within the framework of different classification tasks, the task presented in this study can be described as a multi-class classification problem. This term is used to describe problems where there are more than two classes and every sample must be classified into only one class exclusively. Because

the performance of different models on a multi-class problem is straightforward to evaluate, the current study focuses on this type of experimental setup. The real-world scenario of an acoustic system employed in a greenhouse for insect sound classification must in contrast be described as a multi-label problem. For such a problem every sample has one or multiple correct labels, thereby including scenarios where two or more different insect species are audible at the same time and therefore in the same sound sample.

Following, the two models used to explore the possibilities of the novel dataset are presented. The first model represents a plain baseline solution, introducing a basic VVG16 model trained on spectrograms of a single microphone channel. The second model introduced is a variant of a WaveNet model operating on multi-channel raw audio data.

Spectrogram Model

In recent years, deep learning classifiers built for the task of image classification, have shown very impressive performance even on very difficult image classification tasks. Specifically, various derivatives of CNN models have proven not just very successful, but also very versatile and easy to train. Therefore, it has become the standard approach to make use of image-classifying CNN models for audio classification tasks, by using them to classify time-frequency image representations, so called spectrograms, of audio signals.

Besides the benefit of being able to use potent image classifiers, the generation of spectrograms also compresses the data size. Samples that are smaller in memory size, can be loaded faster and can be processed by smaller models, which in term become faster as well. Additionally, spectrograms offer the benefit of displaying the content of a signal in a graphic manner that allows humans easy and faster understanding of the signal content, in contrast to listening to audio recordings.

In this study the spectrograms were generated using the scipy library [27]. Precisely, a Fast Fourier Transform (FFT) length of 2048 on a Tukey window with shape parameter 0.25 over 1083 values, overlapping by 135 values was used. The presented parameters were the result of a trial and error pre-experiment, evaluated by what seemed to enhance the clarity of the result to the human eye. Given that the insect flight sounds are of very low frequency, a long FFT window was chosen to enhance the frequency resolution for these low frequencies in the resulting spectrogram.

The resulting spectrogram displays the signals frequency distribution starting from zero to a maximum restricted by the Nyquist-Shannon-theorem as half the sampling rate. For the here used 16 kHz sampled data, this results in a sound frequency range of 0 Hz to 8 kHz. To boil down the data input to the frequency range that appears relevant to the problem of insect flight sound classification, the bottom and top of the spectrograms were discarded, leaving a spectrogram displaying the sound frequency range from 50 Hz up to 2000 Hz. Given the aforementioned noise attenuation characteristics of the anechoic box used to record the insect sounds in the anechoic lab, sounds below 50 Hz were found to be contaminated with environmental noises and building vibrations from outside the box. Given the previously described frequency range of the flight sounds, signals above 2000 Hz ought to have little to no relevant information for the classification task. From a signal processing perspective, cutting the upper and lower parts of the spectrogram is equivalent to applying a perfectly steep high and low pass filter to the time domain signal.

The values of this small spectrogram were then transformed to a dB scale. This has been shown to improve classification results and corresponds to a visual improvement in clarity for the human observer. To allow processing of the spectrograms as image data in the deep learning model, the spectrograms then needed to be converted to 8-bit values and to a value range between 0 and 1.

As a baseline model, processing the so-generated spectrograms, a VVG16 model was employed [24]. Dating back to 2014 this model is not the latest and most sophisticated the computer vision landscape has to offer. On the other hand, it has been proven to be a capable, robust and straightforward architecture. As it has been used as a baseline model in various studies exploring different optimizations for deep learning models, its training dynamics are well understood. Therefore, the VVG-16 model seems like a good candidate to get a first impression when stepping into the uncharted territory of a novel dataset from an unexplored domain.

To make use of the VVG16 model for classifying the spectrograms, it had to be slightly modified. As the original model was built to classify 224×224 pixel RGB images, the model input was modified to handle the 250×42 pixel greyscale images. This way, no further reshaping of the input spectrograms was necessary and they could directly be fed into the model. Further, the original model was built to classify the 1000 classes in the ImageNet dataset. To make the model predict the five insect classes, the final layer was replaced with a dense layer of 5 neurons and softmax activation.

Raw Model

Generating a spectrogram from audio data not only leads to a more accessible representation of the frequency content but also results in a loss of information. As the spectrogram is generated by sliding a window across the signal, the time resolution of the spectrogram is limited to this window size. Furthermore, for adequate resolution of low frequencies in the signal, large windows need to be used. As the target signals in this study are of very low frequency, the use of these large windows further contributes to a very low time resolution of the spectrograms used.

To extract information about the direction of arrival of different sounds and enhance or attenuate selected directions, spatial filtering algorithms rely on fine-grained temporal information in the signals [13]. Therefore, the lower temporal resolution of spectrograms is especially of concern, when using multi-channel data. In practice, this means that the benefit of a multi-channel signal over a single-channel signal is very limited when processed in the form of spectrograms.

Furthermore, in the context of deep learning classifiers, there is a strong case to be made for the use of end-to-end models. The parameters of spectrogram generation cannot be optimized by the training process, as this transformation is a step of feature engineering that is outside of the training loop. The generation of a spectrogram has plenty of such parameters and there exist many different variations for transforming and representing the results of the underlying FFT calculation. Because these parameters and choices are outside the training loop, they might be chosen sufficiently well to achieve a working model, but will never be chosen perfectly. This is especially true for novel audio classification tasks, such as acoustic insect recognition. While, e.g. in the field of bird song classification, the performance of different flavours of spectrograms has been investigated [8], such studies are missing for the application examined in this work.

The attempts to classify raw audio data so far have shown two different approaches. First, different models have been proposed, which try to mimic the steps of generating a spectrogram in the first layers of the model architecture by using convolution and pooling layers [21]. The second approach is represented by the WaveNet model [26], which instead of mimicking the windowed transformation from time to frequency domain, tries to operate on the full detailed time domain signal. Originally introduced as a generative model for music and speech synthesis, the WaveNet architecture has since been successfully adapted to various sound classification tasks [16, 17, 30].

As the mathematics behind beamforming algorithms, such as filter and sum algorithms, are very similar to the operations happening in the convolutional layers of a CNN, the idea of "learned beamforming" seems appealing. First experiments with this approach have demonstrated success [6]. The idea, also referred to as NBF has since been further explored. [22] proposed a NBF layer that was adapted in this study.

In detail, the NBF layer used here consisted of four 1D convolutional layers with kernel size 80, stride 4 and 40 filters each. No activation was performed on the output of these layers. The four individually convoluted signals were then added up and fed into a WaveNet classifier model resembling the setup presented by [30] for the task of music artist classification. The WaveNet classifier implementation was based on code available at [20]. To account for the downsampling effect of the strided convolution in the NBF layer, the size of the pooling layers in the top part of the WaveNet classifier was adjusted. Further, "same" padding was used instead of the originally proposed "causal" padding throughout the entire network. In contrast to the approach presented by [22], where the spatial filtering layer was initialized with beamforming-like kernels, all kernels of the model were initialized at random in this study. This approach represents the attempt to build a model capable of effectively processing multi-channel raw audio data by taking advantage of different signal directions using the neural beamforming layer and WaveNet as a very powerful classifier.

2.2.2 Model Training

When training a deep learning model, one has to employ a data pipeline to feed data to the model during training. Depending on the form of the dataset stored on the disc and the details of the training procedure, this pipeline may contain various steps of data transformation or alteration. In the case of this study, this step is preceded by the extraction of the sound samples from the raw long-form recordings and their random selection. Following, the methods used in this study are described in further detail.

Sound sample extraction

To create a dataset from the insect sound recordings made in the anechoic box, the actual sound events needed to be extracted from the mostly silent recordings. To do so in a reproducible and fully automated manner, a custom-written Python program was used. The following paragraph aims at roughly describing the structure of this program. As the length and therefore data size of the extracted samples greatly limits the complexity of model architectures to be explored because of memory constraints, a rather short sample length of 2500 ms was chosen for this study.

The process of segmentation, based on the idea of activity detection on a prefiltered signal, was loosely based on the steps described in [9]. The sound sample extraction program implements the following steps:

1. Loading one of the 14:13 min long tdms-files.
2. Downsampling the signal from 48 kHz to 16 kHz.
3. Picking the loudest channel of the four channels recorded. This is done by simply summing up the squared values of every channel and picking the channel with the maximum squared signal sum. By reducing the signal from four to one channel, the computational complexity of the following steps can be drastically reduced.
4. Prefiltering the signal by passing it through a 4^{th} order Butterworth low pass filter at 1500 Hz and a $30th$ order Butterworth high pass filter at 180 Hz. The prefilter-

ing minimizes the influence of environmental sounds leaking through the anechoic enclosure on the following steps of processing.
5. Windowing the signal and estimating the signal energy within each window. This is done by calculating the signal energy for every window as the sum of the squared signal values. A window size of 3279 values, hopped by 1024 values, was used for this.
6. Activity detection by thresholding the window energies. The threshold for window energy was set to 1.6 times the mean of all the window energy's found in the 14:13 min recording. Parts of the signal that exceed this energy threshold are marked as "activity".
7. Extracting non-overlapping, equal-length sound samples containing one or multiple phases of "activity". For this segmentation process, a cascade of if-else cases is used to divide the signal into samples of 2500 ms in length based on the previously calculated activity markers. Segments shorter than 1 s without any previous or following activity segments within a 2500 ms range are discarded as noise.
8. Saving the extracted sound samples. The final sound sample, stored on disk, contains the original, unfiltered, four-channel signals of the extracted segments at a sampling rate of 16 kHz.

Using this program a total of 46.023 samples were extracted from the 312 h of insect sound recordings made in the anechoic box.

As balanced datasets ensure easy interpretability of the various metrics used for model evaluation, an identical number of 7350 samples was randomly picked from the results of the sample extraction process for each of the insects. This dataset was then split into training, validation and test data using a 60-20-20 split.

To create a corresponding set of noise sound files, the environmental sound recordings made in the greenhouse, as described in Sect. 2.1.4, were simply split into non-overlapping 2500 ms segments and saved as 16 kHz files also. This way a dataset of 1739 environmental sound samples was created. This dataset was then also split into files used during training, validation and testing using a 60-20-20 split.

General training pipeline and hyperparameter choice

The steps of the training data pipeline to feed the datasets stored on the disc to the model were as follows.

1. Loading training and validation insect sound files from the disc.
2. Shuffling insect training data files (and reshuffling every epoch of training).
3. Applying data augment to the training files.

In order to be able to mix insect and environmental sounds, for implementation reasons, the same number of both sound files was needed. As the dataset offers far fewer different environmental sound samples than insect sounds, the next steps necessary were:

4. Building training and validation lists of environmental sound files as big as the number of available insect sounds by repeatedly randomly choosing files from the environmental training sound database or validation training database, respectively.
5. Loading the chosen training and validation environmental sound data.
6. Mixing the environmental sounds with the insect training and validation sounds.
7. Clipping the signals to a value range of -10 to 10, mimicking the behaviour of the A-D-Converter.
8. Prefiltering the signal by:
   - 8.1 Applying a 4^{th} order Butterworth high pass filter at 50 Hz.
   - 8.2 Convert the signal from a value range of -10 to 10 to a value range of -1 to 1.
   - 8.3 Centring the signal around 0 by element-wise subtracting the signal mean.
9. For the spectrogram model, the spectrogram is generated as the last step of the pipeline by:
   - 9.1 Picking the loudest channel, as the channel with the maximum sum of the squared signal.
   - 9.2 Generating the spectrogram as described in Sect. 2.2.1.

The training pipeline as well as the models were implemented using Tensorflow 2.10 [1]. Both models were trained using the Adam optimizer [7], set up with the default values implemented in Tensorflow. The raw and spectrogram models were trained with a learning rate of 9×10^{-5} and 1×10^{-4}, respectively, when trained from scratch. The training was automatically terminated using Early Stopping, with a patience of 10 epochs monitoring the validation loss. The spectrogram model was trained on batches of size 1024 and the raw audio model on batch size 64. The machine used for training employed a NVIDIA GeForce RTX 3090 with 24 GB of RAM for GPU processing.

Data augmentation

The correct and plentiful use of data augmentation has been proven to play a vital role in improving deep learning results. By deliberately randomly varying all parameters of the training data, which are a priori known to not correlate with the classes, one can ensure these parameters will not contribute

to the decisions of the model. Typical augmentation techniques for audio classification tasks [28] in the time domain and their applicability to the task at hand shall be discussed here briefly:

- Pitch shift
- Time stretch
- Time shift
- Random gain
- Signal inversion
- Addition of Gaussian or pink noise
- Addition of background noise

Shifting the pitch slightly on a speech recognition dataset, may help the model learn to deal with speakers that naturally vary in pitch (e.g. male vs. female speakers). For tasks where the pitch of the signal is distinctive to the class, such as bird song classification or the example at hand dealing with insect sound, these augmentations however would lead to a devaluation of the training data.

The same can be said for randomly stretching or compressing the signal in time. While the meaning of a word is not changed regardless of whether it is spoken fast or slow, the signals of insects might contain sensible temporal patterns that could be destroyed by applying this augmentation.

Randomly shifting the signal in time by a small margin, however, is a technique that is applicable to most domains. The meaning of a signal remains the same, independent of its onset in time. Therefore, a random time shift in the range of ± 250 ms was applied to the training signals during augmentation.

Also, a slight variation in signal magnitude is sensible for most audio data, as it correlates with different distances between the sound source and the microphone. High variations in the signal magnitude, however, might conceal a reasonable correlation between sound level and classification, as it is well within reason to assume that bigger insects might be louder than smaller insects. For this reason, a limited random gain in the range of $\pm 20\%$ was applied to the training data during augmentation.

As sound waves are oscillating waves of high and low pressures in the air and therefore create data oscillating between positive and negative values, another common augmentation is signal inversion. By multiplying the signal by -1, the positive and negative values of the data are flipped. As this should not alter the meaning of the signal, this technique was also applied at random during the augmentation step in the training pipeline.

To train models that are more resilient to noise, be it digital noise from the recording hardware used or environmental sounds, Gaussian or pink noise is often added to the signals during training. As the unique proposal of this study is the addition of domain-specific background noises to the training data, such randomly generated noise signals were not used during the data augmentation step in this study. The use of environmental sounds shall be discussed in the following paragraph.

Exposure to environmental noise sounds during training
The previously described augmentation methods are usually applied with very low intensity. Applying more drastic augmentations will lead to the training and validation data becoming too dissimilar and thereby hindering model convergence. Depending on the domain and the specific alterations applied during augmentation, the augmented data does not necessarily perfectly resemble realistic data or may show alteration artefacts. Therefore, data augmentation is usually only applied to the training data and not to the validation or test data.

In the unique case of this study, the addition of the sounds from the greenhouse to the insect sounds recorded in the box should, however, produce very close to realistic data. The simulated signals are likely to only deviate from real signals of insects recorded in the greenhouse in the following two ways. First, a minimal amount of additional background noise is present due to the background sounds that made their way through the walls of the anechoic box during the recording of insects in the acoustic lab. Second, the amount of noise introduced by the different devices in the measurement chain, is doubled in the simulated data, as it is present in the insect recordings as well as in the background recordings that are mixed by simply element wise adding up the two signal arrays.

Nevertheless, mixing the data to simulate greenhouse conditions represents the only chance of building a development environment for an acoustic insect detection system in the greenhouse. Furthermore, mixing the insect and greenhouse sounds in different ratios results in a unique possibility for studying the behaviour of different models when exposed to different noise levels.

By mixing not only the training but also the validation and test data with recordings from the greenhouse, this study crosses the line from augmentation to simulation. Using this simulation approach, the training and validation data no longer become dissimilar when increasing the level of the mixed-in noise sound.

Nonetheless, the proposed models did not converge when exposed to insect sounds mixed with full-scale environmental sounds during training from scratch. Training the same models on the clean insect recordings from the acoustic lab, however, proved unproblematic. As a solution, a fine-tuning approach was applied. By training the models on clean data and then continuing training on noisy data, model convergence was facilitated. Specifically, the two models pre-trained on the clean data were fine-tuned, first, on data mixed with environmental sounds at 10 % of their original

level. The fine-tuned models were then further fine-tuned on data mixed with environmental sounds at 20 % of their original level and then again on data mixed with full-scale environmental sounds.

When exploring this training approach, a search for the optimum learning rate at each noise level revealed that for the raw audio model increasing the learning rate from 9×10^{-5} to 5×10^{-4} for the training steps including noise, showed the best results. For the spectrogram model, no such observations could be made and therefore the learning rate was kept constant at 1×10^{-4} for training at every noise level.

2.2.3 Model Evaluation

As previously described, the unique dataset in this study allows the simulation of different sound levels of realistic environmental noise. This technique not only allows for unique possibilities when training classification models but also for model evaluation.

First of all, every model was evaluated on data representing the conditions it was trained on, by mixing the test data with environmental noises at the same level as was used during model training. Furthermore, because the environmental sound samples are chosen at random from the environmental sound test set, the evaluation metrics fluctuate slightly. Depending on whether a quiet insect sample will be mixed up with a loud or a quiet environmental sound sample, it might be easier or harder to classify. To account for these fluctuations, the evaluation was repeated 10 times, except for the case of evaluation on clean insect sounds. The results presented here are the mean values of these 10 repetitions.

Second, the simulation offers the possibility to further explore the model performance under different noise conditions. Therefore, each model was also evaluated at various noise levels it was not trained at. This proved to offer some interesting insights into how the models adapt to the varying training conditions. Because the models are initialized with random values at the beginning of training, the results vary slightly for each repetition of training. Each model was trained five times on clean data and then subsequently fine-tuned and refine-tuned at increasing noise levels.

3 Results

In total, both models were trained from scratch five times on clean data and then subsequently fine-tuned and refine-tuned on increasingly noisy data. Out of the five attempts to train the raw model on pure laboratory data, four converged. Continuing the training for these four models with increasing noise levels saw one more model diverge. Out of the three remaining, two yielded very good results and one showed a heavy bias to one class at every noise level, leading to much worse results.

In contrast, all five attempts to train the spectrogram model converged towards very similar results. This must be explained by the simpler architecture of the spectrogram model, which, as expected, proved to be easier to train to its full potential.

Following, the results of evaluating these models in the different manners described in Sect. 2.2.3 are presented. Additionally, to foster well-founded explanations of the results presented, an analysis of the sound level distribution in the dataset is introduced.

3.1 Training and Evaluation on Pure Insect Recordings form the Acoustic Lab

To generally assess the capability of the two compared models to recognize different insect sound patterns and at the same time value the level of useful information contained in the insect sound dataset, the results of both models on pure insect sound recordings are introduced. The top two lines of Table 1 on page 15 show the results of evaluating the two models on clean data after training on pure insect sounds. For the spectrogram model the table shows the average of all five training attempts. The results depicted for the raw audio model are the average of the training attempts that converged under the respective noise conditions. On the clean insect

Table 1 Results of evaluating the two model types under noise conditions similar to their training conditions and conditions differing from their training conditions. The values represent the average of the successful training attempts for each model type at the respective noise level

Row	Model type	Training noise level (%)	Evaluation noise level (%)	Accuracy (%)	Precision (%)	Recall (%)
(1)	Spectrograms	0	0	87	89	86
(2)	Raw Audio	0	0	94	95	93
(3)	Spectrograms	0	100	20	20	20
(4)	Raw Audio	0	100	21	21	21
(5)	Spectrograms	100	0	23	34	11
(6)	Raw Audio	100	0	47	66	35
(7)	Spectrograms	100	100	28	94	7
(8)	Raw Audio	100	100	44	87	25

sound data, the raw audio models on average slightly outperformed the baseline spectrogram models with an average classification accuracy of 94 % vs. 87 %.

Further insights into the models behaviours and the dataset are provided by the confusion matrices in Fig. 2 on page 13. This figure details the classification accuracy and the misclassification for every class. It shows the progression of the training attempt that led to the best performance in the last training step, for the spectrogram model on the left and the raw model on the right. The two confusion matrices at the top (Fig. 2.1 and Fig. 2.2) represent the evaluation results of the two models trained and evaluated on pure lab recordings. Both models show a nearly perfect classification of the *A. aphidimyza* (class e), while performance for the other classes lacks behind.

This can be partially explained by a dataset flaw. Unfortunately, the original plan of placing only one insect species inside the recording box at a time, to create an unmistakably labelled dataset, was interfered with by a set of fungus gnats for some of the recordings. These small flies found their way into the acoustic lab via eggs in the potting soil. Closer investigation, after recording, revealed that the recordings made from *B. terrestris* (class a) and *E. balteatus* (class c) contained similar low-level flight sounds that need to be attributed to the presence of fungus gnats inside the insect-rearing cage together with the insects under investigation. Even though subjectively, the sounds of the target species seem to be most dominant in the recordings, regarding the model evaluation, it must be considered that the models might have a valid reason to misclassify any of the two contaminated classes for one another.

Looking at the false classifications in Fig. 2.1 and Fig. 2.2, it can be observed that both models confuse *P. prasina* (class b) and *C. septempunctata* (class d). As the recordings of both of these insects were not contaminated by fungus gnat sounds, the reason for this confusion could be the anatomical resemblance between the two insects. While *C. septempunctata* is considerably smaller than *P. prasina*, these two insects are the only ones in the dataset having a pair of harder cover wings, called the *elytra* in the former and the *hemelytra* in the latter, covering their delicate flight wings underneath. This could cause similar sounds from both insects that stem from folding and unfolding their wings.

3.2 Analysis of Sound Level Distribution in the Datasets

In an attempt to gain further insights into the effects influencing classification results under noisy conditions, the relative sound level distribution of the dataset was analysed. Figure 3 on page 14 shows the distribution of the occurrence of samples of different average sound levels in the training dataset for the five different insects investigated, as well as for the background sounds. The graph was generated by, first, putting the sound samples of the respective training datasets through the data pipeline for model training presented in Sect. 2.2.2. This included high pass filtering and offset correction, but did not include augmentation operations. Second, the root mean square (RMS) value of every sample was calculated as an estimate of its acoustic sound level and plotted as a histogram for every class.

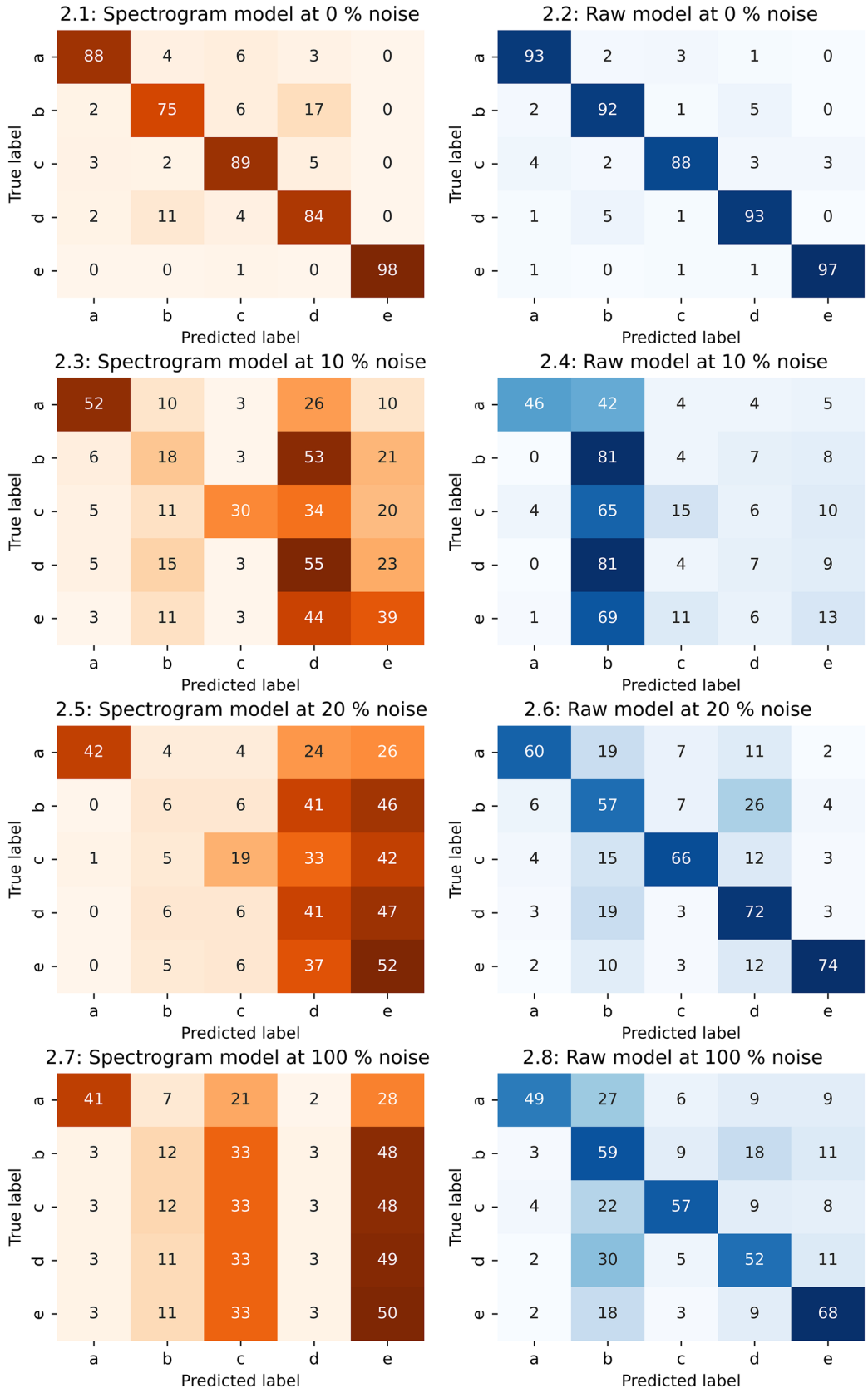
Figure 3 confirms the subjective impression that *B. terrestris* is the loudest of all insects investigated. Further, the distribution of the *B. terrestris* sound levels shows two peaks. To a lesser extent, this can also be observed in the distribution of *C. septempunctata*, *P. prasina* and *E. balteatus* sound levels. These distributions show one peak on the quiet and one on the loud side of the spectrum.

Taking into account that the utilized recording setup allows the recording of very low-level sounds, this could be explained by different types of sounds recorded from the same insect. Reviewing the recorded sounds, it was found that the sound of most insects can broadly be divided into two types. One that clearly must be flight sounds and another that can best be described as non-flight insect movements. As there was no camera in the recording chamber, one can only hypothesise what these non-flight sounds originated from. Listening to the recordings, one comes to think of insects walking on the mesh of their rearing cages or cleaning their wings. These two different types of sounds, the latter of which are much quieter, could explain the two peaks in sound level distribution.

3.3 Model Fine-tuning to Different Noise Levels

Carefully approaching real-world conditions, the two models were fine-tuned on data with increasing simulated background noise levels. The bottom section of Fig. 2 (Fig. 2.3 to 2.8) shows the confusion matrices of the two best models after fine-tuning and refine-tuning on noise levels increasing up to realistic background sound levels using the training procedures explained in Sect. 2.2.2.

It becomes apparent that even the addition of environmental noises downscaled to just 10 % of their realistic level, has a devastating impact on both model performances. Column patterns show in both confusion matrices, suggesting that the models start randomly guessing instead of making meaningful predictions. The only class that can be recognized somewhat safely by both models at this second stage of training is *B. terrestris* (class a). It must however be mentioned that, while the five different training attempts of the spectrogram model showed very consistent results (see Online Resource 3), one of the training attempts of the raw model not shown yielded 59 % average accuracy at 10 % of


2.1: Spectrogram model at 0 % noise
2.2: Raw model at 0 % noise
2.3: Spectrogram model at 10 % noise
2.4: Raw model at 10 % noise
2.5: Spectrogram model at 20 % noise
2.6: Raw model at 20 % noise
2.7: Spectrogram model at 100 % noise
2.8: Raw model at 100 % noise
True label
Predicted label
a
b
c
d
e

◂**Fig. 2** Confusion matrix for the spectrogram-based model (left) and the raw audio model (right), trained and evaluated on pure insect sound (at the top), or fine-tuned and evaluated on simulated greenhouse background noise levels of 10 %, 20 % and 100 % (progressing downwards). Classes are denoted as: **a** *Bombus terrestris*, **b** *Palomena prasina*, **c** *Episyrphus balteatus*, **d** *Coccinella septempunctata* and **e** *Aphidoletes aphidimyza*

noise, instead of 32 % as depicted in Fig. 2. It however later resulted in slightly worse performance in the final training step than that of the model depicted (see Online Resource 4).

Further training on increasing noise levels sees the best of the raw audio model training runs converge to a state that on average correctly classifies 57 % of all sounds under simulated real world noise levels. It clearly outperforms the Spectrogram model, which is stuck guessing.

The mid section of Table 1 show the average results of evaluating the two model types under noise conditions differing from their training conditions. As expected, the models trained on clean data perform very poorly when exposed to environmental background noises, as can be seen in row 3 and 4 in Table 1. What was not to be expected, is that models fine-tuned on noisy data, show a drastic decrease in performance on clean data, shown in row 5 and 6 of Table 1.

This effect is exaggerated in the spectrogram model, which seems to be guessing completely, as indicated by the low recall values (Table 1, row 5). While the raw audio model, too, seems to unlearn a big part of what made it perform so well in the first step of training, it still performs better when there is less noise in the training data (Table 1, row 6 vs. row 8). Accordingly, the fact that the spectrogram model trained on noisy data does perform worse on clean data indicates that instead of finding a new meaningful minimum in the solution space, the repeated fine-tuning of the spectrogram model rather forces it to find differing shallow local minima in the solution space, by guessing one of the classes.

Considering the sound level distribution of the datasets presented in Fig. 3, one might hypothesise the following explanation. As the majority of all insect sounds, but those of the *B. terrestris*, are very low-level sounds, even at only 10 % of realistic noise levels they seem to be masked almost completely by the background sounds.

Intuitively both models should at least be able to identify flight sounds audible to human ears under realistic background noise conditions. Considering the sound level distribution of these audible insects such as *P. prasina*, *E. balteatus* and *C. septempunctata* displayed in Fig. 3, the louder flight sounds, however, appear to be very rare in the dataset. The overwhelming amount of low-level samples, which seem to present an unsolvable task to the models, seems to completely hinder the spectrogram model from converging towards a solution that recognizes at least the loudest flight sound patterns. This analysis suggests that removing the quieter insect sound samples from the dataset could improve the results of the spectrogram model under noisy conditions.

In comparison, the raw audio model seems to be more resilient to the influence of background sounds added by the simulation. During preliminary tests, different versions of the raw audio model were investigated with e.g. different learning rates. The majority of the versions showed results and behaviour very similar to the spectrogram model described here. It is only when trained with the precise setup described in detail in Sect. 2.2.1 that some training runs yield the very good results presented in Fig. 2. While the filters in the NBF layer in the poor performing versions of the raw audio model seem to look for parallel patterns in all four microphone channels, the filters in the good model seem to be triggered by distinct phase-shifted patterns in the different microphones (see Online Resource 5). This indicates that the best versions of the raw audio model can utilize their NBF layer effectively to filter out some of the background noises by using spatial information included in the multichannel audio signals.

What remains very surprising is that Fig. 2 shows that, while the loud *B. terrestris* (class a) is clearly the best recognized class in noisy conditions for the spectrogram model, this is not the case for the raw audio model. The raw audio model in contrast accurately recognizes the smallest and quietest insect, *A. aphidimyza* (class e), with the best accuracy at high levels of simulated background sounds. In part, this could be explained by the fungus gnats contamination in the *B. terrestris* and *E. balteatus* (class c) data. Additionally, as the test in clean laboratory conditions suggested, the, still much louder, sounds of the *P. prasina* (class b) and the *C. septempunctata* (class d) seem to be more similar than the rest of the data, as explained in Sect. 3.1.

Finally, it could be possible that the good detectability of *A. aphidimyza* (class e) observed in Fig. 2 exists not in spite of, but because of the fact that it is the quietest. Because *A. aphidimyza* is so small and quiet, the dataset for this class appears to be almost exclusively containing flight sounds and not a mix of flight and non-flight sounds, as found for the other classes. As flight sounds should be most characteristic to each class, this could ease the recognition of the *A. aphidimyza*. Further, the flight sounds of *A. aphidimyza* are so quiet, that even the sensitive microphones used in this study will only pick them up if they originate close to the microphone. In contrast, the sounds of the louder insects will be picked up by the microphones, regardless of their position in the cage. This could mean that the *A. aphidimyza* data contains a more uniform directional pattern, which could lead to a more effective spatial filtering for this class learned by the NBF layer of the raw audio model.

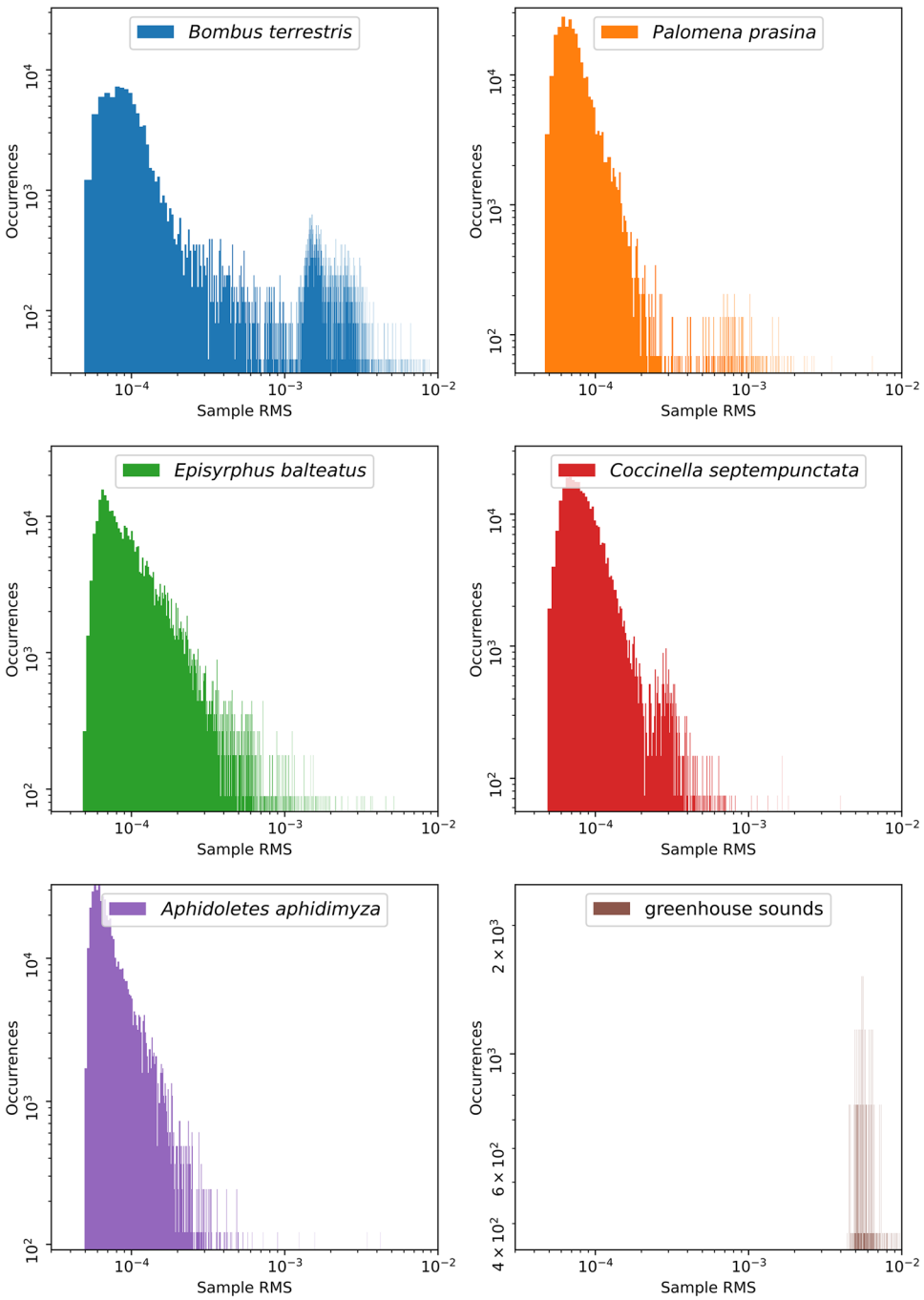


Fig. 3 Histograms with 10.000 bins depicting the number of occurrences of samples with similar RMS values for everyone of the five insect classes in the training dataset and the greenhouse background noise sound samples in the training noise dataset

4 Discussion and Outlook

This study aimed to not only show the feasibility of recognizing insects by their sound but to develop a system that can do so with significant environmental noise present in the recordings.

The training of both models on clean insect sounds showed impressive results. Considering the fungus gnats problem in the dataset, the performance of the raw audio model must be valued as nearly perfect. Previous studies have shown in detail that insect flight sounds differ in their base frequency [3] and that insects can be successfully distinguished using these wing beat patterns [4, 19]. Taking into account the distribution of the sound levels and their possible explanation as consisting of two different sound types, the results could be interpreted as proof that the insects investigated in this study can also be distinguished by every other sound of their body's motion, loud enough

to be recorded. This is an unobvious result, as these very low-level sounds are not only unknown to human ears and can only be heard using special recording hardware, but also subjectively seem to differ very little between classes when played back from the recordings.

To bridge the gap from noise-free laboratory conditions to real world background noise levels, this study proposes repeated fine-tuning training on simulated data with increasing noise levels. The procedure of restarting training on an increasingly difficult modified version of the original dataset is unique to this study. However, there is related work investigating the effect of repeated training restarts on a constant dataset. This work reports on the benefits of such restarts regarding the optimisation process underlying model training [11].

The results under realistic noise conditions reveal the different capabilities of the two models investigated. While the spectrogram model seems to only recognise the very loud flight sound of *B. terrestris* in these challenging conditions, the best raw audio model seems to be able to effectively use the spatial information in the dataset to reduce the masking effect of the background sounds. Surprisingly this even enabled the model to recognize the majority of the *A. aphidimyza* sounds in simulated greenhouse conditions. As this insect is not audible to human ears, this result impressively demonstrates the potential of high quality measurement equipment combined with state of the art pattern recognition algorithms to gather and utilize information that is beyond human perception.

The key difference between the two model architectures was the spatial filtering capacity, only built into the raw audio model via the use of the NBF layer. However, effectively training this NBF layer proved to be little stable, as only two out of five training attempts converged to a solution with a working NBF layer. During training, a more purposeful initialization, as proposed by [22], may have stabilized this convergence towards useful kernels.

Considering the recording setup in the acoustic lab, the spatial information that could be learned from the datasets must be seen as very restricted. As during the insect recordings, all insect sounds originated from within the insect-rearing cage directly in front of the microphone, the model will have learned to listen only to sounds coming from this area. As the noise sounds originate from every possible direction in the space around the microphones, only a small fraction of them should reach the microphones in the same direction of arrival as the insect sounds in this dataset. Even though in real-world scenarios insect sounds might also originate from every possible direction, this solution at least enabled the raw audio model to correctly classify the majority of sounds coming from in front of the microphones, while the non-directional spectrogram model failed on all but the loudest sounds.

The evaluation of the models under noise conditions differing from their training conditions presented a mix of expected and unexpected results. As expected, models exposed to no noise during training failed when exposed to noise during testing. What was surprising was the poor performance of models trained under noisy conditions and tested on clean data. The results again showed better performance of the raw audio model than the spectrogram model. For both models, however, it meant that the fine-tuning training on noisy data caused them to, at least partially, unlearn how to make accurate predictions on clean data.

The limitation of this study is that based on the presented data, there is no way to assess how good the simulated greenhouse recordings are at imitating actual real-world recordings of insects in a greenhouse. Further, the multi-class scenario investigated in this study does differ from real-world conditions, which must be described as a multi-label problem. However, it shall be noted that the transfer from a multi-class problem to a multi-label problem proved unproblematic in the preliminary experiments of this study. In addition, to maximise the number of training samples available, insect sounds recorded on different days were included in the training, validation and test data set. Although it is considered unlikely because the noise-shielding recording environment and high-pass filtering exclude almost all background noise from the insect noise samples, it cannot be ruled out that some models may be able to detect the recording date in some of the noise samples. This could be a potential information leak from training to test data and should be investigated in future work.

This study represents a first step towards closing the gap from high performance in laboratory conditions to a robust solution under real-world noise conditions. It illustrates both the immense potential and the limitations of the application of deep learning in sensor development. While the experiments on the clean insect sound data demonstrate the immense pattern recognition capabilities of modern deep learning models by almost perfectly classifying sounds that mean nothing to human ears, the experiments in simulated noise conditions are a reminder that even the best AI systems are constrained by physical limitations. A signal masked by noise beyond recognition represents an unsolvable pattern recognition task. Impressively, the proposed use of multi-channel signals and spatial filtering has proven to be a tool that can push this limit of unrecognisability beyond human perceptual capabilities. Proceeding to future work the following conclusions can be drawn from this study:

- Recording separate insect and environmental sound datasets to simulate not only real-world, but also intermediate ratios of signal-to-noise mixtures, proved to be fruitful. The possibility to per-train the models on clean data and then fine-tune them on noisy data showed to be the very precondition to attempting this problem, as models trained on realistic noise levels from scratch did

not converge. Further, being able to approach realistic conditions in small steps provided valuable insights into model behaviours. Future work should expand on this approach by increasing the datasets.

- Real-world recordings are necessary to verify that the simulated greenhouse sounds actually represent a good estimation of real insect sounds recorded in a greenhouse.
- The WaveNet model proved to be a capable solution for this acoustic classification in this domain.
- The NBF layer proved to be a key building block of the raw audio models that performed best, justifying the effort of recording and processing multi-channel raw audio data. Spatial filtering remains the only solution to increase the SNR of a signal masked by noise in a similar frequency range. Future studies in this difficult field should further investigate the potential of multi-microphone setups and multi-channel data processing for insect recordings.
- Excluding the impossibly quiet samples from the dataset could decrease the difficulty of the classification under noisy conditions and thereby foster model convergence towards more useful solutions. Regarding future work, this poses the difficult question of drawing the line between samples that are too quiet and those that are just loud enough.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s13218-023-00812-x.

Funding Open Access funding enabled and organized by Projekt DEAL. The presented work was conducted in the context of the research project IPMaide. The project is supported by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support programme.

Data availability The insect sound dataset on which this work is based will be made available as part of a future dataset publication. The code for the raw audio model will also be made available as part of a future publication further investigating this architecture.

References

1. Abadi M, Agarwal A, Barham P, et al (2015) TensorFlow: large-scale machine learning on heterogeneous distributed systems. http://download.tensorflow.org/paper/whitepaper2015.pdf
2. Bennet-Clark HC (1998) Size and scale effects as constraints in insect sound communication. Philos Trans R Soc Lond B Biol Sci 353(1367):407–419. https://doi.org/10.1098/rstb.1998.0219
3. Byrne DN, Buchmann SL, Spangler HG (1988) Relationship between wing loading, wingbeat frequency and body mass in homopterous insects. J Exp Biol 135:9–23
4. Chen Y, Why A, Batista G et al (2014) Flying insect classification with inexpensive sensors. J Insect Behav 27(5):657–677. https://doi.org/10.1007/s10905-014-9454-4
5. He W, Lu L, Zhang B, et al (2019) Spatial attention for far-field speech recognition with deep beamforming neural networks. arxiv:1911.02115v2
6. Hoshen Y, Weiss RJ, Wilson KW (2015) Speech acoustic modeling from raw multichannel wave forms. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 4624–4628
7. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arxiv:1412.6980v9
8. Knight EC, Poo Hernandez S, Bayne EM et al (2020) Pre-processing spectrogram parameters improve the accuracy of bioacoustic classification using convolutional neural networks. Bioacoustics 29(3):337–355. https://doi.org/10.1080/09524622.2019.1606734
9. Le-Qing Z (ed) (2011) Insect sound recognition based on MFCC and PNN. https://doi.org/10.1109/CMSP.2011.100
10. Li B, Sainath TN, Narayanan A, et al (eds) (2017) Acoustic modeling for google home. https://doi.org/10.21437/Interspeech.2017-234. http://www.cs.cmu.edu/~chanwook/MyPapers/b_li_interspeech_2017.pdf
11. Loshchilov I, Hutter F (2016) SGDR: stochastic gradient descent with warm restarts. arxiv:1608.03983v5
12. Mankin RW, Hagstrum DW, Smith MT et al (2011) Perspective and Promise: a century of insect acoustic detection and monitoring. Am Entomol 57:30–44. https://doi.org/10.1093/ae/57.1.30
13. McCowan I (2001) Microphone arrays: a tutorial. http://www.aplu.ch/home/download/microphone_array.pdf
14. Montgomery GA, Belitz MW, Guralnick RP et al (2021) Standards and best practices for monitoring and benchmarking insects. Front Ecol Evol. https://doi.org/10.3389/fevo.2020.579193
15. Njoroge AW, Affognon H, Mutungi C et al (2016) Frequency and time pattern differences in acoustic signals produced by *Prostephanus truncatus* (Horn) (Coleoptera: Bostrichidae) and *Sitophilus zeamais* (Motschulsky) (Coleoptera: Curculionidae) in stored maize. J Stored Prod Res 69:31–40. https://doi.org/10.1016/j.jspr.2016.06.005
16. Oh SL, Jahmunah V, Ooi CP et al (2020) Classification of heart sound signals using a novel deep WaveNet model. Comput Methods Programs Biomed. https://doi.org/10.1016/j.cmpb.2020.105604
17. Pandey SK, Shekhawat HS, Prasanna S (eds) (2019) Emotion Recognition from raw speech using Wavenet, IEEE, Piscataway, NJ. https://doi.org/10.1109/TENCON47323.2019. https://ieeexplore.ieee.org/servlet/opac?punumber=8910516
18. Phung QV, Ahmad I, Habibi D et al (2017) Automated insect detection using acoustic features based on sound generated from insect activities. Acoustics Australia 45(2):445–451. https://doi.org/10.1007/s40857-017-0095-6
19. Potamitis I, Rigakis I, Fysarakis K (2015) Insect biometrics: optoacoustic signal processing and its applications to remote

monitoring of McPhail type traps. PLoS ONE 10(11):e0140,474. https://doi.org/10.1371/journal.pone.0140474
20. Pyeon M (2018) wavenet-classifier. https://github.com/mjpyeon/wavenet-classifier
21. Sainath TN, Weiss RJ, Senior A, et al (eds) (2015) Learning the speech front-end with raw waveform CLDNNs. https://doi.org/10.21437/Interspeech.2015-1
22. Sainath TN, Weiss RJ, Wilson KW, et al (eds) (2016) Factored spatial and spectral multichannel raw waveform CLDNNs. https://doi.org/10.1109/ICASSP.2016.7472644
23. Sainath TN, Weiss RJ, Wilson KW, et al (2017) Raw multichannel processing using deep neural networks. In: Watanabe S, Delcroix M, Metze F, et al (eds) New era for robust speech recognition. Springer International Publishing, Cham, pp 105–133. https://doi.org/10.1007/978-3-319-64680-0_5
24. Simonyan K, Zisserman A (2014) very deep convolutional networks for large-scale image recognition. arxiv:1409.1556v6
25. Sotavalta O (1947) The flight-tone (wing stroke frequency) of insects. Acta Entomol Fenn 4:1–117
26. van den Oord A, Dieleman S, Zen H, et al (2016) WaveNet: a generative model for raw audio. arxiv:1609.03499v2
27. Virtanen P, Gommers R, Oliphant TE et al (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17(3):261–272. https://doi.org/10.1038/s41592-019-0686-2
28. Wei S, Zou S, Liao F et al (2020) A comparison on data augmentation methods based on deep learning for audio classification. J Phys: Confer Ser 1453(1):012,085. https://doi.org/10.1088/1742-6596/1453/1/012085
29. Weik MH (2000) telephone frequency. In: Weik MH (eds) Computer science and communications dictionary. Kluwer, Boston. https://doi.org/10.1007/1-4020-0613-6_19249
30. Zhang X, Gao Y, Yu Y, et al (2020) Music artist classification with WaveNet classifier for raw waveform audio data. arxiv:2004.04371v1