Automated Bat Call Classification using Deep Convolutional Neural Networks

Preprint · April 2021

CITATIONS
0

READS
2,180

6 authors, including:

Karl-Heinz Frommolt
Museum für Naturkunde - Leibniz Institute for Research on Evolution and Biodiversity
47 PUBLICATIONS **916** CITATIONS

SEE PROFILE

Automated Bat Call Classification using Deep Convolutional Neural Networks

E. Schwab[a], S. Pogrebnoj[a], M. Freund[a], F. Flossmann[a], S. Vogl[b], K.-H. Frommolt[c]

[a]Deggendorf Institute of Technology, Deggendorf, Germany;
[b]University of Applied Sciences Munich, Munich, Germany;
[c]Museum für Naturkunde, Leibniz-Institut für Evolutions- und Biodiversitätsforschung, Berlin, Germany

ARTICLE HISTORY
Compiled April 16, 2021

ABSTRACT
Identification of bats is most practically done by exploiting the characteristic features of their echolocation calls. This usually involves expert knowledge, expensive equipment and time-consuming post processing of previously recorded calls. Automated solutions exist, but are usually not as accurate as human experts. We present an automated solution for the processing of bat calls and identification of bat species with extremely high classification accuracy that can be used during life recording or in an automated post processing software. Our algorithm is the first application of a Deep Convolutional Neural Network to classify bat species based on sound spectrogram images of their echolocation calls.

We tested several deep CNN architectures including a modified Google Inception and a ResNet50 architecture. The nets were trained on a very large call database consisting of images of snippets of call spectrograms. All our software was developed in the Python programming language and is available on request.

KEYWORDS
bat calls, echolocation, neural network, image classification

1. Introduction

Nearly all European bat species are endangered and are usually under more or less strict monitoring. Their numbers are declining and some species have already vanished from former habitats. The reasons for the decline are, amongst others, the loss of habitats, roost and hibernation sites, the decline in insect populations, the use of agrichemicals or the fatalities caused by wind turbines which interfere with seasonal migration, mating and hunting patterns (Dietz & Kiefer 2018).
Since bats are very sensitive to changes in environment the diversity and abundance of this group can be used as bioindicator (Jones et al. 2009). Therefore, bats are also considered in most environmental impact assessments. This is usually done by evaluating automated recordings from bat recorders, devices that record sound also in the ultrasonic frequency range above 20 kHz. These recorders are often automatically triggered so that they record only short sound snippets (a few seconds) whenever a

F. Flossmann. Email: florian.flossmann@th-deg.de
S. Vogl. Email: stefanie.vogl@hm.edu
K.-H. Frommolt Email: karl-heinz.frommolt@mfn-berlin.de

sound event in the ultrasound region meets certain trigger criteria.
These sound events can be the characteristic echolocation calls of bats or their social calls, but also the sounds katydids and other insects or small mammals. Beside animal vocalisations even technical such as pulses from electrical pasture fences can trigger the recording (Middleton 2020). In addition to these complications there is a wide range of recording devices (ultrasonic microphones and related hardware), so that the characteristics of the collected data may be technically different (Walters et al., 2013).
However, all the collected sound snippets have to be analyzed to scan for the occurrence of bat echolocation calls. These echolocation patterns vary often strongly between species and can even change with the hunting situation and the topography (Walters et al., 2013). Human experts therefore distinguish most bat species by their echolocation calls, often even just by listening to calls that are transposed into an audible frequency range (accomplished by slowing the signals down by a factor of ten) (Barataud 2020).
The identification becomes more accurate when a graphical representation of the calls in the form of a spectrogram' is evaluated, a plot that shows the evolution of frequency over time in the calls (figure 2). These spectrograms show characteristic patterns that are easily distinguished from interference even by the inexperienced observer and are, for the expert, a valuable tool for the identification of bat species.
Since the manual evaluation of sound recordings is extremely time-consuming, attempts are made to evaluate these spectrogram patterns automatically by means of computer software. Standard methods for automated classification use characteristic features of the calls such as mean duration, peak frequency or mean frequency that are derived from the audio signals (Armitage & Ober 2010; Parsons & Jones 2000). These classification algorithms are based on a reliable segmentation of calls (call-no-call identification) and automated tracking of the fundamental frequency of the echolocation call (Obrist et al., 2018). The results from automated procedures are reliable and deliver results that are comparable to those delivered by human observers in terms of detection quality (Jennings et al., 2008). However, if a very reliable species identification is needed, it is recommended to check the spectrograms manually (Russo & Voigt 2016).
The most promising and up to date approach for the segmentation (detection) of calls uses spectrograms from labeled sound signals together with deep learning algorithms to detect the calls (Mac Aodha et al., 2018). The advantage is mainly that it is not necessary to extract synthetic predefined features such as peak frequency from a sound snippet but let the Neural Network work with the complete information i.e. the spectrogram. This approach is very promising not only for call - no call decisions but also for classification of specific species. We therefore apply different architectures of Convolutional Neural Networks (CNNs) with spectrogram images as input. CNNs are the most powerful machine learning algorithms and state of the art for image classification (Krizhevsky et al. 2012; Khan et al. 2020).

To build up even deeper network structures that are able to extract features on different scales the invention of the so called inception layers and residual learning were of crucial importance (Szegedy et al. 2015, 2016, 2017; He et al., 2016). These networks are alleviating problems during the training phase of the network due to the vanishing-gradient problem, which is getting more and more important the deeper the network is. In addition to that the propagation of features within the network structure is substantially improved, resulting in an increased classification quality when the network is applied to unknown test data. In this study we compare the performance

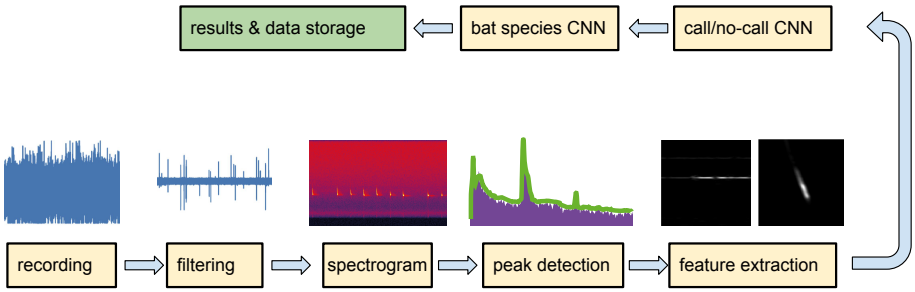

Figure 1. Flow chart of the Bat classification algorithm

of standard CNNs with increasing deepness as well as the Inception-v1 and ResNet-50 architectures (Szegedy et al. 2015; He et al., 2016).

2. The Software

2.1. Concept

The fundamental idea of our software is to exploit the advantages of the spectrographic representation of bat calls where characteristic features such as frequency range, mean frequency or frequency variation of the call become much more prominent. With these spectrogram *images* (fig 3) we train an image processing Artificial Neural Network. As a spectrogram of a complete call sequence can have - at reasonable resolution - hundreds of thousands of pixels, which would slow down the training and classification process for the network significantly, we restrict ourself to images of *single calls*, into which we divide the call sequences.
As state-of-the-art in image classification are the so called 'Deep Convolutional Neural Networks' we use different architectures from this model class (see 2.4).
The program consists of the following steps (see fig. 1):

(1) the signal is recorded or a previously recorded signal is read
(2) the signal is filtered by a band-pass filter ($15 - 120\,\mathrm{kHz}$)
(3) the spectrogram of the signal is computed
(4) a simple algorithm searches for peaks in the spectrogram that could possibly be bat calls
(5) each possible call is surrounded by a 10 ms window and cut out
(6) all possible calls in the spectrogram window are fed into a first CNN, designed and trained to only distinguish bat calls from environmental noise
(7) bat calls that are recognized as such are then fed into a second, much larger Deep CNN, that is designed and trained to distinguish 18 different bat species occurring in Germany
(8) all classification results for the analyzed spectrogram window are summed up

The program, written in Python, exists in two different implementations: as a user-operated bat recorder software to work together with an ultrasonic microphone on a laptop or tablet, and as a fully automatized desktop solution to process large, previously recorded wav-files from unsupervised bat recorders.

2.2. Components

The core of the software - the classificator - is able to accept either realtime sound files or sound files that contain signals slowed down by a factor of ten, both with arbitrary sample rate and 16bit resolution. From sample rate and stretch factor (1x or 10x), the time and frequency scales are calculated. Depending on the implementation, we use as input stream either a microphone channel or a continuously read data file.

2.3. Preprocessing

The image-classifying Deep CNNs are relatively robust against noisy or deformed input signals. The reason is that the networks itself contain various sublayers that act as filters by compressing and decompressing or convoluting the signals. In the case of ultrasonic recordings this means that crackling, noise, crickets and frogs, deformed or disturbed signals, and other disturbances that might thwart other approaches like, for instance, feature extracting algorithms, do not, as we found, substantially impair the classification results.
We therefore abstained from too complicated pre-filters in favour of unaltered signals. The only filter we use is a simple band-pass filter with cutoff-frequencies at 15 kHz and 120 kHz, respectivley. This filter cuts out disturbances in the audible region and makes for a clearer spectrogram (see figure 2 (a), (b) and (c)).
The first step is to fourier transform the signal piecewise, because we search, with a self developed algorithm, for peaks that could possibly be bat calls in the frequency domain. For this first step, we work on the unaltered signal, the sampling rate is here that of the recording or microphone, respectively. The window length is here always 10 ms, a hanning window is used for the FFT.
The above mentioned band-pass filter is applied simply by omitting the respective frequencies. Figure 2 (c) shows the search window. Inside this window, columns of the spectrogram are analyzed separately for the characteristic peaks in the envelope of the fourier spectrum that appear when a bat call is recorded. Figures 2 (d) and (e) show the corresponding plots.
If a possible call is found, it is placed in the midst of a 10 ms window and cut out (figure 2 (f)). In order to obtain signals comparable to those used for the training of the CNN, the possible call snippets are then time-expanded and resampled to 44 kHz. The snippet is then transformed into a spectrogram using a moving hanning window to obtain a greyscale-image of 137x137 pixels (see fig. 3). These spectrogram thumbnails of 'suspicious' sound events are then handed over to a first CNN.

2.4. Classification with different CNN Architectures

To develop a reliable algorithm for the detection of calls and their classification for the most common german bat species, a large dataset of labelled bat echolocation calls is required (Foody et al., 1995). We used a dataset created by R. Skiba for a monography of European bat calls (Skiba 2003) which is now freely available from the archive of animal sound recordings at the Museum für Naturkunde in Berlin (www.tierstimmenarchiv.de). At the moment, our database consists of approximately 45000 single calls extracted from 1500 recordings including 18 resident german species. This data is sufficient to train call-no-call identification Neural Nets and the subsequent species classification network as well as for a subsequent evaluation of the results.

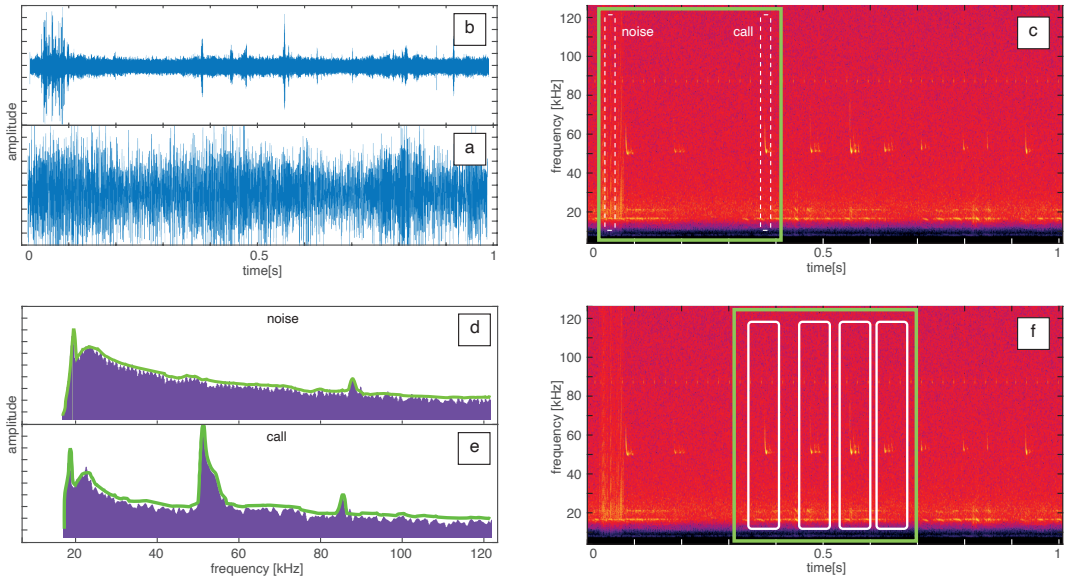

Figure 2. Signal processing in the software. Unfiltered bat call in time-amplitude representation (a). Same call, but high-pass filtered (b). The same call in a spectrogram (c). Inside the green box potential calls are located by searching for peaks along the frequency axis (dotted white slots). Bat call (e) and noise (d) from (c) plotted along the frequency axis. Windows around potential calls that will be cut out and processed by the first neural network.

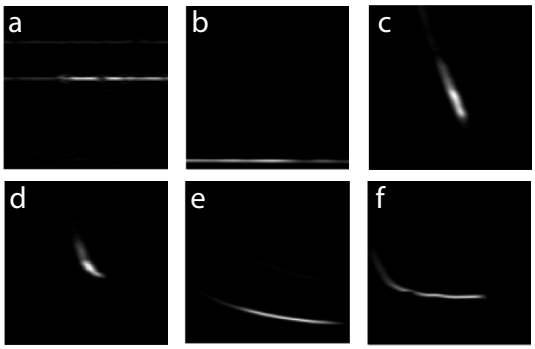

Figure 3. Examples of greyscale spectrograms of sound events that are handed over to the Neural Networks. Spectrograms of: noise event (a), nyctalus noctulae (b), myotis emarginatus (c), pipistrellus pygmaeus (d), eptesicus serotinus (e) and pipistrellus pipistrellus (f).

Depending on the application (real time classification with a bat detector or post processing of previously recorded data) two fundamentally different approaches for the processing of incoming signals are possible. The two different designs are shown in figure 4.

- **single stage algorithm**: the incoming signal is presented to a neural network that has a 19 dimensional target vector. This target vector includes the 18 represented bat species and in addition to that an additional class for incoming noise. In that procedure the network learns to separate noise from bat calls simultaneously. This approach is advantageous insofar as only one single network has to be trained and stored for recall. The workflow is simple and the computational cost for training and recall is comparably low with high classification accuracy. This is a good basis for real time classification with portable devices.

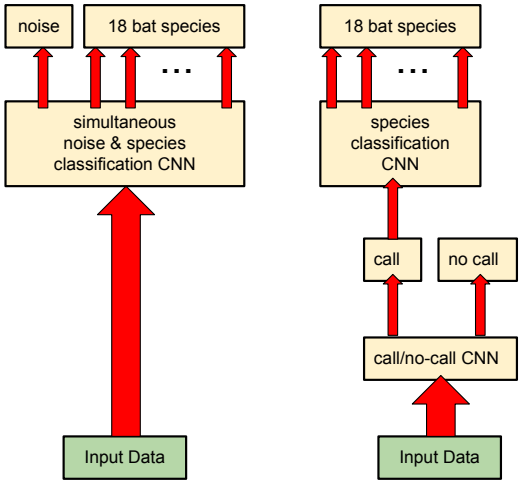

Figure 4. Single stage algorithm (left) and dual stage algorithm (right)

- **dual stage algorithm**: the incoming signal is first passing an identification network which decides if the snippet contains noise or potential bat calls; if the snippet is classified as noise it is rejected. In that case no further action is taken. As soon as a bat call is identified the signal is fed into to a second neural network, the actual classification network, where 18 bat species are discriminated. In this case the topologies of the identification net and the classification net are the same, only the target vector is different. The first network has only two targets, call or noise, while the second has an 18 dimensional target vector corresponding to the number of included species. This method of course involves the training of two separate Neural Nets which leads to an increase in computational time and memory for the storage of the trained nets compared to the single stage algorithm. The time for the recall of a trained net is also slightly extended. However, the two step algorithm has advantages in terms of slightly increased classification accuracy. Thus the method is suitable whenever computational resources are not too restricted such as for the post processing of large datasets (bdAnalyzer).

For both approaches four different Neural Network topologies are tested.

- CNN6: 6 Convolution layers with 3 intermittent Pooling layers and a stack of 4

fully connected layers
- CNN10: 10 Convolution Layers with 5 intermittent Pooling layers and a stack of 4 fully connected layers
- CNNIncept: modified version of the 'Inception-v1' Net developed by Google (Szegedy et al. 2015). Two Convolution and Pooling blocks, followed by a stack of 9 inception layers build the feature extractor. The final classification part with the derived feature vector as input is consisting in 4 fully connected layers. Overall, this architecture is 22 layers deep.
- CNNResNet: modified version of the 'ResNet-50' architecture, a 50 layer Deep CNN developed by Microsoft. (He et al., 2016). The architecture comprises one Convolution and one a subsequent MaxPooling Layer followed by 49 Convolution Layers. Average Pooling is applied before the feature vector is fed into the fully connected layer, activated with a softmax function for final classification. The shortcut connections are included between Convolution Layer two and fifty and skip three Layers respectively.

The first two architectures are adapted, classical versions of deep CNNs. The third and the fourth allow for an even deeper network topology as the incorporation of inception layers reduces the number of free parameters substantially while shortcut connections strengthen information flow through the net. Both mechanisms lead to increasing classification accuracy whenever the network is applied to unknown data (Szegedy et al. 2016; He et al., 2016). The computational effort is kept acceptable even if the network has increased complexity (deepness) of topology.

2.5. Implementation

We have, at the moment, two different software implementations of our algorithm that fit our needs, but are willing to adapt the software to the requirements of field workers and researchers.

- Bat Detector Software:
  The first version is a GUI based bat detection software, originally intended for use on a RaspberryPi microcomputer equipped with a touchscreen and an ultrasound microphone. Meanwhile, it is mainly used on a small Windows10 touchscreen tablet computer , for the ease of file handling, battery management and computing power. As we use Python, the software is easily portable and runs on Windows, MacOS and Linux. If a touchscreen is available, no mouse or keyboard is needed for operation.
  The software works, in principle, with any (driverless) USB microphone. We tested it with devices from Dodotronic, Petterson and Wildlife Acoustics.
  The software shows a live spectrogram window; in the background, all recorded sound is written into a continuously growing temporary file, regardless of any user action on the GUI of the software. The recording only stops when the software is closed, so that the whole bat detection session can be re-analyzed afterwards on a desktop computer and so that the bat detector can also operate in unsupervised mode.
  On button press, a region of interest can be selected and an automated call classification can be performed.

Table 1. Number of bat calls in the database

Species	count
Barbastella barbastellus	327
Eptesicus nilssonii	1245
Eptesicus serotinus	2681
Myotis bechsteinii	329
Myotis brandtii/Myotis mystacinus	2393
Myotis dasycneme	2142
Myotis daubentonii	6074
Myotis emarginatus	849
Myotis myotis	2366
Myotis nattereri	3040
Nyctalus leisleri	2884
Nyctalus noctula	3005
Pipistrellus kuhlii	4619
Pipistrellus nathusii	4695
Pipistrellus pipistrellus	7885
Pipistrellus pygmaeus	566
Plecotus austriacus/auritus	411
Vespertilio murinus	1495
Noise	4713

- Desktop Analysis Software:
  The second implementation of our classification algorithm is a GUI software intended to be used on a desktop computer for the fully automated analysis of previously recorded sound files. The software is called the "bdAnalyzer" - big data analyzer - for its ability to operate on wav-files of (principally) unlimited size.
  The sound files can be real-time recordings or slowed down by a factor of 10, of arbitrary sample rate. **The results of the analysis will the parts of the sound file that contain identified call sequences and spectrogram images of the respective parts, together with a list of the classification results and their relative positions in the file.??**

In each case, the result of the classification for one single call is a list of 18 probabilities for the call to belong to one of the 18 different species in our database. These lists, one for each call in the analysis window, are summed up and the three species with the highest score are given as result.
This may seem confusing in comparison to the established method - where a complete call sequence is analyzed in total - but has the advantage that recordings where call sequences of different species overlap can be analyzed equally simple.

3. Results

3.1. Evaluation of different network topologies on the database

Most of the 18 classes were represented equally well in the dataset (see table 1). Only Barbastella barbastellus with 327 calls, Myotis bechsteinii with 329 and Plecotus austriacus/auritus with 411 calls were underrepresented. Very common species such as Pipistrellus pipistrellus or Myotis daubentonii also appear more frequently within the data.

The CNNs were trained on 80% of the available calls that were randomly selected.

Table 2. Classification accuracy of the network architectures for the 18 included species on the test set (four different CNN topologies).

	CNN Type			
Species	CNN6	CNN10	CNNIncept	CNNResNet
Barbastella barbastellus	93.41	95.36	99.85	100.0
Eptesicus nilssonii	79.97	82.84	92.93	96.21
Eptesicus serotinus	64.59	67.77	77.38	93.14
Myotis bechsteinii	89.91	92.84	97.94	100.0
Myotis brandtii/Myotis mystacinus	88.61	91.89	96.64	96.19
Myotis dasycneme	11.47	22.48	57.98	93.21
Myotis daubentonii	69.25	76.80	75.93	97.73
Myotis emarginatus	94.94	97.92	99.94	98.29
Myotis myotis	89.58	92.92	97,92	99.57
Myotis nattereri	94.78	97.82	99.91	98.11
Nyctalus leisleri	74.79	80.74	90,02	94.62
Nyctalus noctula	94.85	97.97	99,95	97.93
Pipistrellus kuhlii	78.76	80.41	92.20	92.97
Pipistrellus nathusii	79.61	82.69	92.59	88.90
Pipistrellus pipistrellus	88.95	92.01	97.66	98.54
Pipistrellus pygmaeus	94.91	97.92	99.97	98.10
Plecotus austriacus/auritus	85.25	90.00	96.30	97.83
Vespertilio murinus	79.21	82.70	92.73	93.31
Mean Accuracy	**80.71**	**84.62**	**92.10**	**96.13**

The networks for both the single stage and the dual stage algorithm were trained and tested on the remaining 20% of the data that was unknown to the Neural Networks. The results of the validation can be seen in Table 2.
In general the model with the lowest complexity and deepness, the CNN6, showed the lowest classification accuracy. In average 80.71 % of the calls were classified correctly. An extension of the model with more convolution layers added improves the results to 84.62%. Using the CNNIncept, the overall classification performance rises to 92.10%. The best results were shown by the deepest CNN with shortcut connections, the CNNResNet model, where 50 Convolution layers are included. In that case 96.13% of the calls were identified correctly. In all cases noise was identified with 100% accuracy.

Even if some species are underepresented in the dataset the CNNs classify them accurately. Barbastella barbastellus and Myotis bechsteinii are classified with 100 % and Plecotus austriacus/auritus with 97.83 % hit rate by the CNNResNet. Thus imbalanced classes are not problematic in terms of accuracy. The species with the lowest accuracy i.e. values between 92% and 93% are represented very well in the data. The reason for misclassification is therefore not missing data but the fact that the calls of certain species are very similiar and thus hard to distinguish even by experienced human experts. An example is Myotis dasycneme (93.21 %) which can easily be taken for Myotis daubentonii.

3.2. Dependence of the validation accuracy on the database size

As neural networks are data-driven models a sufficiently large database is the key for good performance, independent of the network architecture. To assess if the size of our database is appropriate, we trained the CNNResNet with different portions of our database: 5%, 20%, 40%, 60% and 80% of the available calls were used in different training runs. The network was, in each case, trained for 50 epochs. In each case the same unused 20% of the database were used for evaluation and the model with the highest validation accuracy was selected.

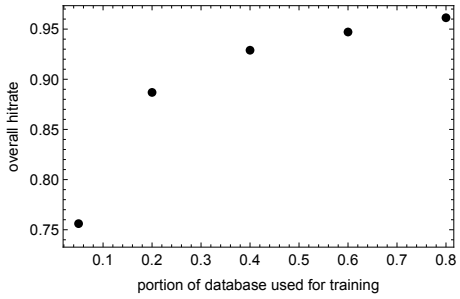

Figure 5. Overall hit rates of the CNNResNet depending on the portion of the database used for training.

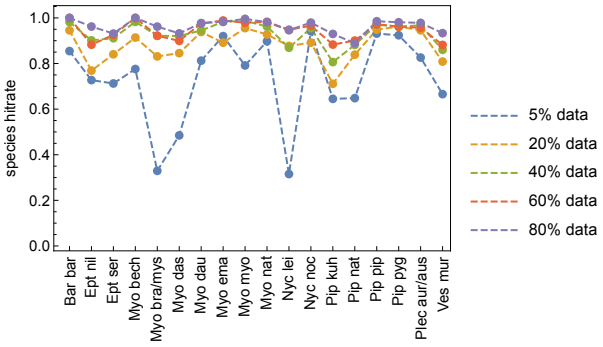

Figure 6. Hit rates for the different bat species depending on the portion of the database used for training.

As expected, the larger the database used for training, the better were the validation results (see fig 5). With only 5% of the database used for training, the overall hit rate is a mere 76% in comparison to the 96% reached when 80% of the calls were used.

If the hit rates are broke down by species, the classification accuracy is found to be relatively equal amongst the different species as soon as at least 40% of our database are used. In our case this means approx. 18000 single calls for 18 different species or roughly 1000 single calls per species!

However, for a very small training set the accuracy for specific species such as Myotis brandtii/mystacinus or Nyctalus leisleri is reduced drastically. To assess the classification accuracy of the CNNs in detail, we calculated the so-called confusion matrices for the evaluation of the networks. Here, not only the percentages of correctly classified calls for each species are listed, but it is also shown onto which species a call is most probably misclassified. Figure 7 shows the confusion matrices for the case of the CNNResNet, trained on 5% (left) and on 80% of the database (right). The numbers in the matrices are color coded for the plot, in the diagonal are the numbers from table 2. Interestingly, but not surprisingly, the species that are most easily confused because of their close relationship can be found as colored clusters in the confusion matrices: myotis dasycneme and daubentonii form a confusion cluster, for example, or the closely related pipistrellus nathusii and kuhlii. Also the nyctalus species noctula and leisleri cluster, as the two very similar eptesicus bats nilsonii and serotinus, respectively. Thus the remaining misclassifications are not stemming from shortcomings in network architecture or training but are due to similar input patterns that are generally hard to distinguish. In a manner of speaking the network makes the same mistakes as human experts would.

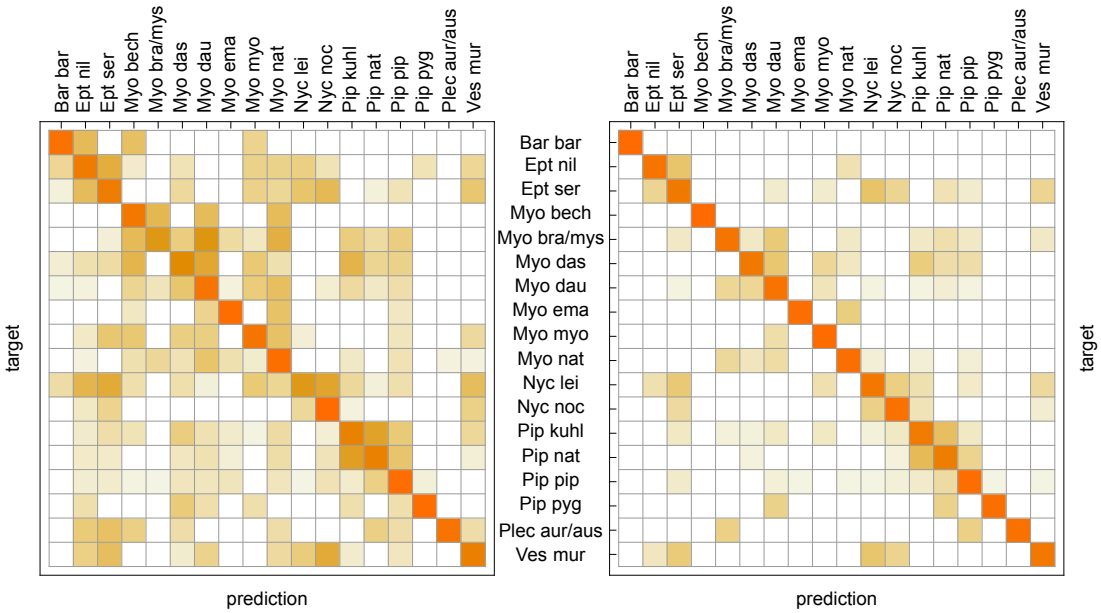

Figure 7. Confusion matrices for the CNNResNet trained on only 5% of the database (left) and trained on 80% of the database.

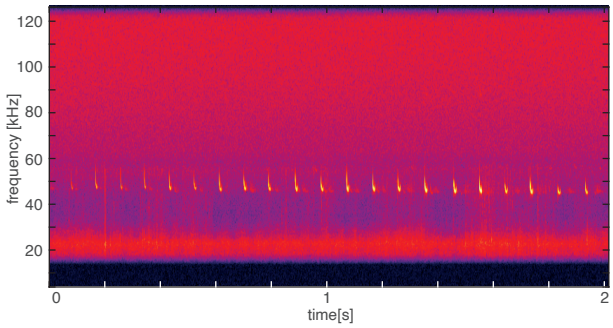

Figure 8. Call sequence of a pipistrelle bat, an example of a very clear recording. The software decides for Pipistrellus pipistrellus.

3.3. Performance in the field

Finally, we want to show some typical examples for the performance of the algorithm in real-live situations.

The first example is a relatively clear recording of a pipistrelle bat (in this case, an individuum personally known to the authors as it is an annual summer guest), with an underlying floor of cricket sounds and some weak double and triple echoes (fig. 8). The software detects all calls and some of the louder echoes and decides with a probability of nearly 100% for Pipistrellus pipstrellus.

The second example is the call sequence of a whiskered bat (either Myotis mystacinus or Myotis brandtii, we do not try discern between the two species as their calls are nearly indistinguishable). The sequence is interspersed with social calls, most probably of the pipistrelle bat from the above example (fig 9). Thanks to the classification being on single call basis, the social calls can be ignored by the algorithm (as the species database does not contain labelled social calls, we classify them as 'noise') and the calls of the whiskered bat are correctly identified. The third example is an exceptionally

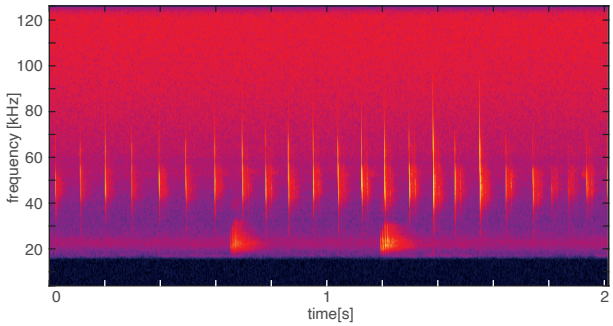

Figure 9. Call sequence of a whiskered bat interspersed with social calls of a pipistrelle bat. The software ignores the social calls and decides correctly for a whiskered bat.

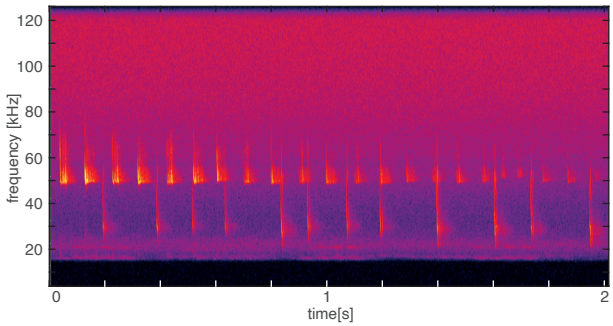

Figure 10. Two overlapping call sequences, one of a pipistrelle bat (Pipistrellus pipistrellus), the other one (lower frequencies) of a long eared bat (Plecotus). The software decides correctly for both species with approximately equal probabilities.

difficult one: a heavily disturbed call sequence of a serotine bat (probably) (fig 11). The interferences are very loud crickets, the higher harmonics of the bat calls which appear due to an overdriven microphone and several noise bands at constant frequencies which were caused by the badly shielded electronics of the recording Raspberry Pi, its power supply and its touch screen. Despite all the interferences, the algorithm decides nevertheless for Eptesicus serotinus, although with a much lower probability of around 50%.
The last example shows a recording that includes two different species with overlapping call sequences (fig 10). Again, thanks to the software working on single calls, the two species - a pipistrelle bat (Pipistrellus pipistrellus) and a long eared bat (Plecotus auritus or austriacus, we do not discern between the two as their calls are too similar) - are both correctly identified. The classification probabilities for the two species are in this case approximately equal, depending on how many calls of each species are identified within the analyzed window.

Also worth mentioning is the fact that the last two recordings were made with a different microphone, easily discernable by the different look of the spectrogram. The performance of the algorithm is, of course, sensitive to different microphones, as the database was collected with different equipment than we use now, but the classification nevertheless also works with high accuracy. To counteract for this, we have designed a GUI software tool that can be used to extend the existing call database with own classified recordings and train the CNN with this extended database. The additional training allows the net to learn the characteristics of specific microphones and improves

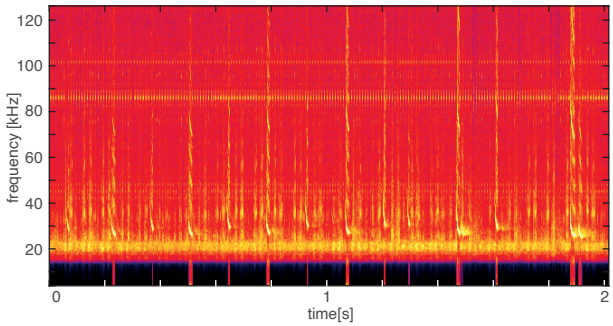

Figure 11. Heavily disturbed call sequence of a serotine bat with interference caused by loud crickets and badly shielded electronics.

the performance of the classification substantially.

4. Discussion

We have developed a method to exploit the capabilities of Deep Convolutional Networks that are usually deployed for image classification on the field of automated bat call classification. Using spectrograms as graphical representation of the audio signals, we trained state-of-the-art CNNs on a large database of labeled single echolocation calls.
With this approach, we achieved with an optimal network architecture (Inception or ResNet50 Deep CNN) a classification accuracy on the validation set of far more than 90%, in the case of the ResNet50 architecture up to 96%.
We found the Deep CNNs to be very robust against disturbances from environmental or technical noise, such that high classification accuracies can still be achieved in the field, even with recording equipment that is different from the one originally used for the calls in the database.
To our knowledge, this is, at the moment, the algorithm most promising for the reliable classification of bat calls on a species level, and we can therefore strongly recommend the use and further development of Deep CNNs for the field of bat call classification.
At the moment, we work on further improvements on the stability and speed of our algorithm on 'real data' by working on the preprocessing part of the software; that is, we try to achieve a faster and more reliable detection of potential call events by different peak detection approaches, we explore the possibility and benefit of noise reduction before peak detection and other, minor improvements. The next step will be to build a real time version of the bat detection software that will then be able to automatically detect and simultaneously classify a sequence of echo location calls as soon as it enters the spectrogram window.
Furthermore, despite the algorithm working on a single call basis, we plan to incorporate the information included in additional features of call sequences into our classification algorithm, such as, for instance, call repetition rate and number of calls in a sequence. Even information about the environment - time of day and year, actual outside temperature, geo data like GPS-position and all that could be retrieved from maps, such as orography, height and surrounding vegetation or buildings, could be exploited to further improve the classification accuracy on closely related species such as Myotis brandtii and mystacinus, the Plecotus bats or the Pipistrelle bats kuhlii and

nathusii.
In conclusion, the method of bat call classification based on spectrogram images is most promising and we are confident that this will become the prevalent approach in near future, as it can benefit from the rapid developments in the field of machine learning we are observing at the moment.

References

Armitage, D. W., & Ober, H. K. (2010). A comparison of supervised learning techniques in the classification of bat echolocation calls. Ecological Informatics, 5(6), 465-473.

Barataud, M. (2020). Acoustic ecology of European bats: species identification, study of their habitats and foraging behaviour. Paris, Inventaires & biodiversité series Biotope - Muséum national d' Histoire naturelle.

Botto Nuñez, G., Lemus, G., Muñoz Wolf, M., Rodales, A. L., González, E. M., & Crisci, C. (2018). First artificial intelligence algorithm for identification of bat species in Uruguay. Ecological Informatics.

Dietz, C., & Kiefer, A. (2018). Bats of Britain and Europe, Bloomsbury Publishing.

Foody, G. M., McCulloch, M. B., & Yates, W. B. (1995). The effect of training set size and composition on artificial neural network classification. International Journal of Remote Sensing, 16(9), 1707-1723

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Jennings, N., Parsons, S., & Pocock, M. J. O. (2008). Human vs. machine: identification of bat species from their echolocation calls by humans and by artificial neural networks. Canadian Journal of Zoology, 86(5), 371-377.

Jones, G., Jacobs, D., Kunz, T. H., & Racey, P. (2009). Carpe noctem: The importance of bats as bioindicators. Endangered Species Research 8: 93-115.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, 53(8), 5455-5516.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105. ISO 690

Mac Aodha, O., Gibb, R., Barlow, K. E., Browning, E., Firman, M., Freeman, R., ... & Jones, K. E. (2018). Bat detective. Deep learning tools for bat acoustic signal detection. PLoS computational biology, 14(3), e1005995

Middleton, N. (2020). Is that a bat? A guide to non-bat sounds encountered during bat surveys. Exeter, Pelagic Publishing.

Obrist, M. K. & Boesch, R. (2018). BatScope manages acoustic recordings, analyses calls, and classifies bat species automatically. Canadian Journal of Zoology, 96(9), 939-954.

Parsons, S., and Jones, G. (2000). Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. Journal of experimental biology, 203(17), 2641-2656.

Parsons, S. (2001). Identification of New Zealand bats (Chalinolobus tuberculatus and Mystacina tuberculata) in flight from analysis of echolocation calls by artificial neural networks. Journal of Zoology, 253(4), 447-456.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29,(9), 2352-2449.

Russo, D., & Voigt, C. (2016). The use of automated identification of bat echolocation calls in acoustic monitoring: A cautionary note for a sound analysis. Ecological Indicators, 66, 598-602.

Skiba, R. (2003). Europäische Fledermäuse. Westarp Wissenschaften, Hohenwarsleben.

Stathopoulos, V., Zamora-Gutierrez, V., Jones, K. E., & Girolami, M. (2018). Bat echolocation call identification for biodiversity monitoring: a probabilistic approach. Journal of the Royal Statistical Society: Series C (Applied Statistics), 67(1), 165-183.

Stowell, D., & Plumbley, M. D. (2014). Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. PeerJ, 2, e488.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 770-777).

Walters, C. L., Collen, A., Lucas, T., Mroz, K., Sayer, C. A., & Jones, K. E. (2013). Challenges of using bioacoustics to globally monitor bats. In Bat evolution, ecology, and conservation (pp. 479-499). Springer, New York, NY.