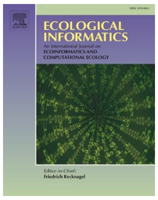
Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



Passive acoustic monitoring of animal populations with transfer learning

Emmanuel Dufourq [a,b,c,*], Carly Batist [d], Ruben Foquet [e], Ian Durbach [f,g]

[a] *Stellenbosch University, South Africa*
[b] *African Institute for Mathematical Sciences, South Africa*
[c] *National Institute for Theoretical and Computational Sciences, South Africa*
[d] *The Graduate Center, City University of New York, USA*
[e] *Biodiversity Inventory for Conservation, Belgium*
[f] *Centre for Research into Ecological and Environmental Modelling, University of St Andrews, UK*
[g] *Centre for Statistics in Ecology, the Environment, and Conservation, University of Cape Town, South Africa*

ARTICLE INFO

Keywords:
Transfer learning
Convolutional neural networks
Deep learning
Vocalisation classification
Bioacoustics

ABSTRACT

Progress in deep learning, more specifically in using convolutional neural networks (CNNs) for the creation of classification models, has been tremendous in recent years. Within bioacoustics research, there has been a large number of recent studies that use CNNs. Designing CNN architectures from scratch is non-trivial and requires knowledge of machine learning. Furthermore, hyper-parameter tuning associated with CNNs is extremely time consuming and requires expensive hardware. In this paper we assess whether it is possible to build good bioacoustic classifiers by adapting and re-using existing CNNs pre-trained on the ImageNet dataset – instead of designing them from scratch, a strategy known as transfer learning that has proved highly successful in other domains. This study is a first attempt to conduct a large-scale investigation on how transfer learning can be used for passive acoustic monitoring (PAM), to simplify the implementation of CNNs and the design decisions when creating them, and to remove time consuming hyper-parameter tuning phases. We compare 12 modern CNN architectures across 4 passive acoustic datasets that target calls of the Hainan gibbon *Nomascus hainanus*, the critically endangered black-and-white ruffed lemur *Varecia variegata*, the vulnerable Thyolo alethe *Chamaetylas choloensis*, and the Pin-tailed whydah *Vidua macroura*. We focus our work on data scarcity issues by training PAM binary classification models very small datasets, with as few as 25 verified examples. Our findings reveal that transfer learning can result in up to 82% F1 score while keeping CNN implementation details to a minimum, thus rendering this approach accessible, easier to design, and speeding up further vocalisation annotations to create PAM robust models.

1. Introduction

Passive acoustic monitoring (PAM) surveys often result in vast quantities of audio data which researchers frequently process manually when searching for particular vocalisation events. Studies have shown that this manual processing can be alleviated through the use of convolutional neural networks (CNNs) (LeCun et al., 1989, 1998) that can be trained to find vocalisation events in large audio datasets (e.g. whales (Bermant et al., 2019), cats (Nanni et al., 2020), birds (Zhong et al., 2021), bats (Paumen et al., 2021), gibbons (Dufourq et al., 2021), chimpanzees (Anders et al., 2021) and seals (Escobar-Amado et al., 2022)). While these results are encouraging several issues still persist, of which we highlight three. Firstly, non-machine learning experts are faced with the decision of deciding on suitable neural network architectures, a non-trivial task. While architectures along with their software implementation do exist (e.g. VGG16 (Simonyan and Zisserman, 2014)), there are no clear guidelines as to which one is suitable for the creation of a PAM classifier. Thus, the development of CNNs is accessible primarily to individuals who have knowledge in deep learning. The second issue is that of training CNNs on acoustic datasets which contain very few vocalisation examples of the various species which can result in overfitting (Hawkins, 2004); a term which means that the model is able to perform well on the training data but poorly on new data that was not used in training. In certain cases it might be difficult to obtain additional audio data (e.g. due to inaccessible habitat or small population size due to species being threatened). Thirdly, hyper-parameter tuning is a time

[*] Corresponding author at: Stellenbosch University, South Africa.
E-mail address: edufourq@gmail.com (E. Dufourq).

https://doi.org/10.1016/j.ecoinf.2022.101688
Received 28 March 2022; Received in revised form 16 May 2022; Accepted 19 May 2022
Available online 14 June 2022

consuming step which involves exploring various neural network hyper-parameters to obtain the most optimal model. When combined, these three problems render the creation and utilisation of CNNs to solve bioacoustics research questions challenging to machine learning experts and non-experts alike.

Similar issues are commonly addressed in the broader deep learning literature by using transfer learning (Pan and Yang, 2009; Weiss et al., 2016), a modelling strategy that uses a model trained on one dataset for the purposes of prediction on another dataset. Transfer learning exploits the fact that parameter estimation in neural networks is performed by iterative numerical optimization, so that parameters found to perform well on one task (say one with input variables **X** and output variables **Y**) can be used as starting points for other, similar tasks (e.g. one with inputs **V** and outputs **W**). Typically the inputs of the target task **V** need to be transformed into a similar form as those of the source task **X**. The neural network which is trained on data (**X**, **Y**) is typically referred to as the pre-trained model.

CNNs, like all neural networks, are made up of layers, with the output of one layer forming the input to the next. Layers can be thought of as organised into two parts: some layers (typically the earlier ones) learn which visual features are relevant to the problem at hand, while others learn how to use those visual features to classify images into one of the target classes. These roles are referred to as feature extraction and classification respectively. All layers contain weight parameters that are simultaneously optimized across the network, reflecting the complementary nature of feature extraction and classification tasks. However, it is also true that the same kinds of visual features (e.g. edges, simple geometric shapes) are important for many image classification problems, and thus a popular approach within computer vision research is to retain only the feature extraction layers of a CNN, removing the last (or last few) classification layers that have been trained for a specific problem and replacing these with new one(s) that can be retrained for the new task (based on the number of categories to be classified within target task **W**). Weights for the new layers must be learned "from scratch", but weights for the feature extraction layers can either be set ("frozen") to their existing ("pre-trained") values, or can be retrained using the pre-trained values as starting values, a process known as "fine tuning" because, if the same visual features are relevant to the new problem, then weights will not change very much from their pre-trained values. The benefit of freezing the feature extraction layers is computational – there are often far fewer weights that require optimisation – which is particularly useful if the target task must be addressed with a relatively small dataset because the risk of overfitting without transfer learning is substantial. Within computer vision research, transfer learning has shown great success over CNNs trained from randomly initialised weights (Tan et al., 2018; Lu et al., 2015; Shao et al., 2018; Jaramillo et al., 2018; Mehra et al., 2018; Lopez et al., 2017). Transfer learning has successfully been applied in other areas of research, e.g. quality prediction (Liu et al., 2019) and medical applications (Yi and Wang, 2021; Kübra Karaca et al., 2021).

Transfer learning within the context of PAM is still relatively underexplored and is the focus of our work and contributions. This study assesses the ability of a variety of modern transfer learning models to accurately classify animal calls within four bioacoustic datasets, and the influence of various modelling decisions – how many calls are annotated, what CNN architecture is used, and preprocessing steps – on that accuracy. We propose that transfer learning can simplify CNN architecture design as it is simpler to load a pre-trained model than to build a suitable CNN. Furthermore, transfer learning is known to alleviate the problem of overfitting and finally, less hyper-parameter tuning would be required. This serves the rationale of our investigation, to explore a large number of pre-trained CNNs in as many configurations as possible to provide a guide to researchers so as to reduce the time spent on the model development and to accelerate the rate at which vocalisation events are found. Initially in an acoustic survey, only a relatively small number of manually annotated vocalisation examples may be available. Thus, we hypothesise that transfer learning would enable further similar examples to be obtained rapidly, which would in turn result in additional model training to find the next set of similar examples at a rapid pace. In particular, we focus on very small datasets, with as few as 25, 50, 100 or 200 examples.

We contribute to PAM by demonstrating that pre-trained CNNs can successfully be trained on very few examples. We compared twelve modern pre-trained CNNs to guide researchers on which are most optimal. We argue that this approach is simpler to implement, and a larger audience of researchers could use this approach rather than implementing CNNs from scratch which requires expert knowledge. Our study is the first to conduct a thorough investigation into transfer learning for PAM and our findings can guide researchers working in PAM. We provide approximately 90 hours of manually verified audio data to train binary classification models. In the next section we discuss relevant literature and argue that there has been no study that has explored transfer learning at such a large-scale within animal bioacoustics.

2. Related literature

We begin by presenting relevant literature on transfer learning for PAM and then make an argument that further research within these two combined areas is still required. The literature reveals that, on average, only one pre-trained CNN was used and that a ResNet based network was a common choice, in particular ResNet50 (He et al., 2016a) was frequently used (Zhong et al., 2020; Henri and Mungloo-Dilmohamud, 2021; Waddell et al., 2021; Efremova et al., 2019; Sankupellay and Konovalov, 2018). Authors mentioned that this was due to its efficiency and accuracy. Sankupellay and Konovalov (2018) used ResNet50 which was pre-trained on the ImageNet dataset (Deng et al., 2009). The authors applied it to a bird vocalisation dataset that contained 46 species. The only modification to the network was that they replaced the last fully connected layer (which was pre-trained on the 1,000 class ImageNet dataset) with 46 softmax units. The spectrograms were duplicated to meet the input of ResNet50 which expects 3 channels. Zhong et al. (2020) compared VGG16 that was randomly initialised to a ResNet50 model that was pre-trained on ImageNet. In both cases, a colour mel spectrogram was input into the network. The spectrograms were resized to match the network's input of 224 by 224. Their models were applied to bird and amphibian vocalisations. ResNet50 pre-trained on ImageNet was also used by LeBien et al. (2020) whereby the pre-trained feature extractor was used and then two fully connected layers were added to the CNN. Zhong et al. (2021) applied ResNet50 to a birdsong dataset that contained three classes (two bird presence and one absence). The CNN was pre-trained on ImageNet and the fully connected layer, followed by a dropout and an output layer was added to the CNN that was then fine-tuned on the birdsong dataset. Colour spectrograms stored as PNG images were used as input to the CNN. Low resource computational devices were the focus of the study of Disabato et al. (2021). Three layers (the first convolution, first pooling, and second convolution) were extracted from ResNet18 (He et al., 2016a) (pre-trained on ImageNet) and a fully connected output layer was added. This model was less computationally expensive compared to commonly used CNNs used within the literature, and thus, sets the premise for exploring models that are both accurate and are able to be executed on hardware with limited resources.

While ResNet based architectures was most commonly used, other architectures were also used within the literature and are presented next. Lu et al. (2021) used AlexNet (Krizhevsky et al., 2012) that was pre-trained on the ImageNet dataset. The input was colour spectrogram images. They explored the effect of varying the number of last layers (starting from the output layer) within the network which were trained from random initialisation. The number of layers that they explored to be randomly initialised were 3, 6 and 9. They created a binary classification model (presence and absence) for which there was little effect on the number of randomly initialised layers. A three class classification model (one for each species of killer whale, long-finned pilot whale and harp seals) was explored and the accuracy increased when a larger number of layers was randomly initialised (from 3 to 9). The training time did not significantly increase as a result. Ntalampiras et al. (2021) created a classifier for cat vocalisations and consequently created a mobile application. The authors used YAMNet [1] pre-trained on the AudioSet-YouTube corpus (Gemmeke et al., 2017) which is an audio event classification dataset containing 512 classes. This CNN architecture is based on VGGish (Hershey et al., 2017) but however it contains fewer trainable parameters which would be advantageous when creating a mobile application. A pre-trained VGGish model was used by Çoban et al. (2020) to soundscape classification. Khalighifar et al. (2022) used Inception v3 (Szegedy et al., 2016) pre-trained on ImageNet to develop a classifier for the monitoring of mosquito populations via smartphone recordings of the wingbeats. The same model was used for the classification of 41 Philippine frog species in the study of Khalighifar et al. (2021). Models pre-trained on ImageNet was common within the literature. Leroux et al. (2021) explored a CNN that was pre-trained on human speech and was used as a feature extractor. Incze et al. (2018) explored MobileNet (Howard et al., 2017) pre-trained on the ImageNet dataset on bird vocalisations collected from Xeno-canto.[2] They fine-tuned MobileNet on three dataset configurations (2, 10 and 50 classes) and two input configurations grey scale and "jet" colour. Both input configurations convert the spectrograms into a different representation where the latter is one that normalises the values between 0 (blue) and 1 (red). The findings reveal that the "jet" colour map produced better results than the greyscaled colour map.

There were not a large number of studies that compared various pre-trained neural network architectures. Xie et al. (2018) compared various implementations of VGG16 pre-trained on the ImageNet dataset. Their results show that VGG16 with transfer learning (whereby the feature extractor was frozen) did not outperform VGG16 with random initialisation. To overcome this, a multi-channel model was created which inputs Short-time Fourier transform, mel-spectrogram and chirplet spectrograms into three separate pre-trained VGG16 models respectively. This approach led to the best results and had considerably less neural network trainable parameters compared to the single VGG16 with random initialisation. Henri and Mungloo-Dilmohamud (2021) collected and created a birdsong dataset from Xeno-Canto to create a CNN classifier. The authors compared MobileNetV2 (Sandler et al., 2018), Inception v2 (Szegedy et al., 2016), ResNet50 and a custom model - the former three networks were pre-trained on ImageNet. Mel spectrograms were used as input to the CNN. MobileNetV2 performed the best and outperformed the custom model by roughly 2 percent. Thakur et al. (2019) compared variants of VGG based models and the findings reveal that when transfer learning was used the model outperformed its counterpart which did not use transfer learning.

There were only two studies for which transfer learning did not improve results as opposed to training the CNN from randomly initialised weights (Morgan and Braasch, 2021; Pamula et al., 2020). There are also studies in the literature for which the authors implement their own CNN from randomly initialised weights and do not make use of transfer learning (Ruff et al., 2020; Dufourq et al., 2021; Nolasco et al., 2019) and other studies for which the authors make use of existing architectures but did not use transfer learning (Bergler et al., 2019; Jiang et al., 2019). To the best of our knowledge, all recent and relevant studies were surveyed.

Based on the literature review, it is clear that transfer learning can result in better PAM classifiers. Furthermore, a thorough analysis has yet to be conducted by comparing a large number of pre-trained CNNs across multiple datasets as a means of guiding researchers. Finally, a comparison of the use of fine-tuning and the effects of the dataset size has yet to be conducted, thus answering the question, "how many audio examples are required to train a CNN model that will perform sufficiently well?" This forms the rationale for this study; a thorough analysis of various CNNs on all possible configurations of transfer learning on four different species. Given that one of the rationales for using transfer learning is to overcome the limitations of overfitting due to small dataset sizes, we explore various dataset sizes to determine which configuration is best suited when creating a PAM classifier in a setting where little data is available. This study compares the performance of CNNs pre-trained on ImageNet on four binary bioacoustic datasets for the vocalisation detection of the critically endangered Hainan gibbon *Nomascus hainanus*, the critically endangered black-and-white ruffed lemur *Varecia variegata*, the vulnerable Thyolo alethe *Chamaetylas choloensis*, and the Pin-tailed whydah *Vidua macroura*. Given the recent work in exploring low resource devices by Disabato et al. (2021), this study will also demonstrate how transfer learning can be used to train CNNs with fewer neural network parameters and thus enable researchers to train on less expensive hardware.

3. Materials and methods

3.1. Data collection

In this study we used four datasets, containing different species, that were collected and provided by various researchers. Several experiments, defined in the next section, were conducted to assess the use of transfer learning in the process of creating binary classification models. In each case PAM devices were set to record for a number of hours across multiple days, however the start time and duration of each recording differed for each study. Details for each dataset are listed below.

1. The *"lemurs"* dataset contained approximately 75 hours of audio data that contained calls of the Critically Endangered black-and-white ruffed lemurs. These were obtained from 4 acoustic monitors that were placed in a sub-humid rainforest site (Mangevo) in the southeast of Ranomafana National Park in Madagascar. Specially, 2 SongMeter SM4's (Wildlife Acoustics) with a sampling rate of 48,000Hz and 2 Swift recorders (Cornell Center for Conservation Bioacoustics) with a sampling rate of 16,000Hz were used. Recordings were collected continuously between May and August 2019. The target detection was the roar-shriek.
2. The *"alethe"* dataset contained approximately 29 hours of audio that contained calls of the Vulnerable Thyolo Alethe. The audio data was collected in the Mount Mulanje Biosphere Reserve, Malawi using 10 Audiomoths (Hill et al., 2019). The sampling rate was set to 32,000Hz and the recordings were obtained over five days in November 2020. The target detection was the single syllable call.
3. The *"gibbons"* dataset contained approximately 70 hours of audio obtained from an existing study on the Critically Endangered Hainan gibbons, whereby 8 Song Meter SM3 recorders (Wildlife Acoustics, Maynard, Massachusetts) were used to collect audio data in Bawangling National Nature Reserve, Hainan, China. The sampling rate was 9,600Hz. While a larger collection of audio recordings exists (Dufourq et al., 2021), we randomly selected and manually annotated 69 files that were not annotated in the original study (Dufourq et al., 2021). This was done to contribute additional annotated data related to Hainan gibbons. Recordings were collected between March to August 2016. The target detection were all vocalisations (phrases and duets).
4. The *"whydah"* dataset contained approximately 60 hours of audio data containing calls of the pin-tailed whydah and was collected using one Audiomoth at the Intaka Island Nature Reserve in Cape Town, South Africa as part of this study. The sampling rate was set to

[1] https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

[2] https://xeno-canto.org/

48,000Hz and these were obtained over two weeks in January 2021. The target detection were individual phrases.

3.2. Data analysis

We treated each dataset as a binary classification problem. During the annotation process we thus annotated each call of the particular species for that dataset as the presence class, and we annotated any other sound (biophony, geophony or anthropophony) as the absence class. For each dataset, every audio file was manually annotated using Sonic Visualiser. This was done using the "boxes layer" feature and enabled us to draw bounding boxes around each call. By doing so, the start and end times were captured along with a binary label (presence or absence). The length of each bounding box varied based on the duration of the sound being annotated. To label the absence class, we randomly placed bounding boxes of varying length throughout each file. Annotating vocalisations of other species was an important consideration as our preliminary results revealed that the neural networks would produce a large number of false positives if other species' vocalisations were not annotated. We only created bounding boxes for the absence class if the sound was within the frequency range of the species of interest for that particular dataset. Thus, if the frequency of the sound was outside of the range for the particular species for the dataset, then that sound was not annotated as it would not benefit the absence class.

Since CNNs require a fixed input size, we studied the vocalisations within the presence class for each dataset to determine the characteristics of the calls which would allow us to create fixed input, which we refer to as segments. For example, the Hainan gibbon calls vary from 2 to 9 s (Dufourq et al., 2021), and thus a suitable input size was 4 s to ensure that the smallest call would fit within the segment. A longer input (>4 seconds) would result in CNNs with more network parameters, an undesirable consequence as this would increase the chances to overfit. A shorter input (<4 s) would not contain enough information, especially in cases where the individual pulses that make up a call are long. A short input could omit parts of the call. Preliminary experiments were conducted on the different datasets to minimise the input length as much as possible, thus minimising network parameters. The characteristics of interest were the call duration, as well as the minimum and maximum values associated with the fundamental frequency for the calls. We did not consider any harmonics within the calls as preliminary results revealed that it was sufficient to only consider the fundamental frequency.

To create the input for the CNNs we performed four pre-processing steps; similar to the approach used in Dufourq et al. (2021). Firstly, we applied a low pass filter on each audio file. This was done as a means of reducing aliasing artefacts which can arise when downsampling an audio file. The cut-off rate associated with the filter was different for each dataset and was selected based on the maximum frequency of the respective species' call within the presence class. Secondly, we downsampled each audio file as a means of reducing the computational requirements for processing all of the files as higher frequencies were not needed. We set the nyquist rate to the maximum frequency for each species' call and set the downsampling rate to be twice the nyquist rate.

Thirdly, we extracted a number of audio segments from each training audio file based on the annotations for both classes. The length, *l*, of the audio segments (denoted in seconds) was different for each dataset. This was done using a sliding window approach. Each annotation contains a start and end time (denoted in seconds). For each annotation, we start by placing the window at the *start time* and extract a segment of audio containing the amplitude values between the *start time* and *start time* + *l*. Then, the window is moved by one second in time and another segment is extracted (*start time* + *1*, *start time* + *l* + *1*). This is repeated a number of times until the end of the sliding window exceeds the *end time* for that annotation. This process is repeated for each annotation, and as a result, a dataset is created containing various audio segments for the presence and absence class.

Table 1
Pre-processing hyper-parameters for each dataset and the number of testing files used.

	Lemurs	Alethe	Gibbons	Whydah
Low pass filter cut off	4000	3100	2000	9000
Downsampling rate	9600	6400	4800	18400
Nyquist rate	4800	3200	2400	9200
Segment duration	4	2	4	3
Number of testing files	46	27	22	78
Testing time (min)	1840	810	1300	1560

Finally, audio segments were then converted into mel-frequency spectrograms representing a two dimensional array. Table 1 presents the spectrogram parameters used in this study. These were determined by studying the characteristics of the calls in each dataset and via preliminary experiments and a validation set. The values associated with the low pass filter, downsampling and Nyquist rate were set based on the maximum frequency for each species of interest. The segment duration were determined by studying the length of the pulses within each vocalisation for the species. Fig. 1 illustrates the pre-processing steps.

3.3. Experimental design

We used four experiments to investigate the effects of transfer learning, especially within the context of data scarcity. More specifically, we attempt to answer the question "can a CNN be successfully trained on very few verified calls?". By answering this, this would enable bioacoustics researchers to spend less effort in manually labelling calls prior to training CNNs. Typically, a researcher would need to manually label a large quantity of data – a labour intensive and time consuming task. We randomly selected a subset of 25, 50, 100, and 200 spectrograms within the presence class. In order to overcome class imbalance issues we randomly augmented the presence spectrograms by applying a time-shift operation to generate enough synthetic spectrograms such that the number of spectrograms in both classes were equal. The time-shifting operation involved taking the starting time of a spectrogram and shifting the data by random integer increment and wrapping back so that the spectrogram duration was not changed. This is similar to how it was implemented in Dufourq et al. (2021) – establishing class balance results in the best performance.

We illustrate this explanation with an example on the *gibbon* data. Once the gibbon binary spectrogram dataset was created, we randomly selected 25 presence spectrograms. There were 3000 absence spectrograms for the gibbons dataset. Thus, to ensure class balance, the 25 presence spectrograms had to be augmented 120 times using time-shifting ($25 \times 120 = 3000$). Thus, we would have 3000 presence and 3000 absence spectrograms for the gibbons. However, as another example, should the sample size be 50, then for the gibbon data we would need to augment 60 times to ensure class balance ($50 \times 60 = 3000$). We set the maximum sample size to 200 as we already know from the literature that CNNs perform well on large datasets, however the scope of this study is on very small datasets. The number of absence spectrograms in the *gibbons* dataset was 3000, for the *lemurs* it was 2500, for the *alethe* it was 1200 and for the *whydah* dataset it was 4100.

All the CNNs in this study were initialised to pre-trained ImageNet weights as this was the most common approach reported in the literature surveyed. We compared 12 popular CNN architectures (Table 2). For each one we removed the classifier and created a randomly initialised (using Xavier initialisation (Glorot and Bengio, 2010)) two-unit softmax output layer. This was done as it is the simplest possible implementation and relates to the goals of this study in keeping the details simple, and to facilitate understanding and usage. We compared two main approaches for transfer learning. The first was to freeze the feature extractor and fine-tune the output layer (denoted as *no fine-tuning*), and the second was to fine-tune both the feature extractor and the output layer (denoted as

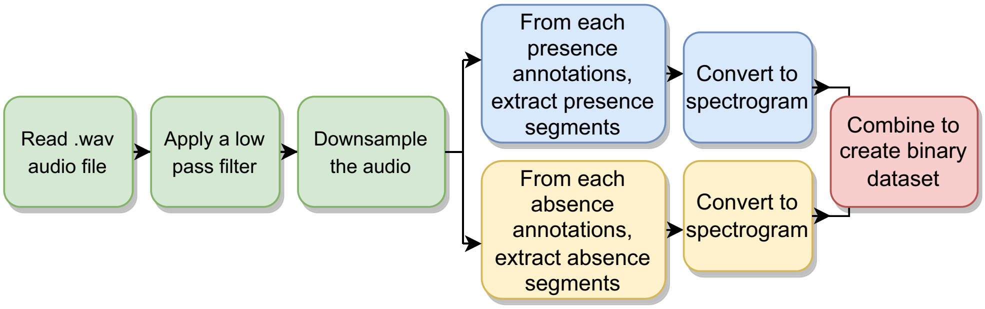

Read .wav audio file
Apply a low pass filter
Downsample the audio
From each presence annotations, extract presence segments
Convert to spectrogram
From each absence annotations, extract absence segments
Convert to spectrogram
Combine to create binary dataset

Fig. 1. Four binary datasets were created. For each one, the dataset was made by reading in every audio file, one at a time, and applying four pre-processing steps. Segment duration varied for each dataset as well was spectrogram parameters, detailed in Table 1.

Table 2
The 12 CNNs compared in this study. The number of trainable network parameters are shown for the case where the feature extractor was fine-tuned (*with FT*) and where the feature extractor was frozen (*without FT*).

Architecture	Study	Parameters with FT	Parameters without FT
DenseNet121	Huang et al. (2017)	6,986,626	32,770
DenseNet169	Huang et al. (2017)	12,537,730	53,250
DenseNet201	Huang et al. (2017)	18,154,370	61,442
InceptionResNetV2	Szegedy et al. (2017)	54,294,626	18,434
InceptionV3	Szegedy et al. (2016)	21,792,930	24,578
MobileNetV2	Howard et al. (2017)	2,275,074	51,202
ResNet101	He et al. (2016a)	42,634,754	81,922
ResNet101V2	He et al. (2016a)	42,610,818	81,922
ResNet152V2	He et al. (2016a)	58,269,826	81,922
ResNet50V2	He et al. (2016a)	23,601,282	81,922
VGG16	Simonyan and Zisserman (2014)	14,731,074	16,386
Xception	Chollet (2017)	20,888,874	81,922

fine-tuning).

For each of the experiments we executed the training of each CNN a number of times so that a distribution of accuracy metrics could be determined. This was also done since there is a stochastic aspect to CNNs and thus reporting on few executions would be misleading. For each execution the weights in the output layer of the CNN were randomly initialised. We split each dataset into training (60%) and testing (40%) by randomly selecting entire audio files – similar to other machine learning studies. To ensure a fair evaluation, we split the data in such a way that the training audio files were mutually exclusive to the testing audio files. The testing files were generally recordings over different days. For testing, the CNNs predicted two softmax outputs on each entire testing file. The final class was assigned based on the softmax output which had a value greater than 0.5. This decision threshold was not optimised to keep the resulting models as accessible and easy-to-use as possible. This process was done using a sliding window of constant time duration. The window is shifted by 1 second until the network had predicted on the entire file. The testing files were manually annotated and thus we could compute the performance of the network on each testing file and report on the F1-score as it was commonly used in literature. Unless stated otherwise, model training and testing was done on Microsoft Azure using the Data Science Virtual Machine and a NCv2-series virtual machine (NVIDIA Tesla P100 GPU).

Given that pre-trained networks expect a 3 channel input (corresponding to the 3 channels in a colour image), the first experiment was an investigation on the type of input spectrogram representation. It was unclear from the literature as to the best representation. We explored the simplest representation for which the spectrogram is a greyscaled image that is duplicated twice to form the 3 channel input (e.g. Sankupellay and Konovalov (2018)) – denoted as *duplicate*. Next, we explored a

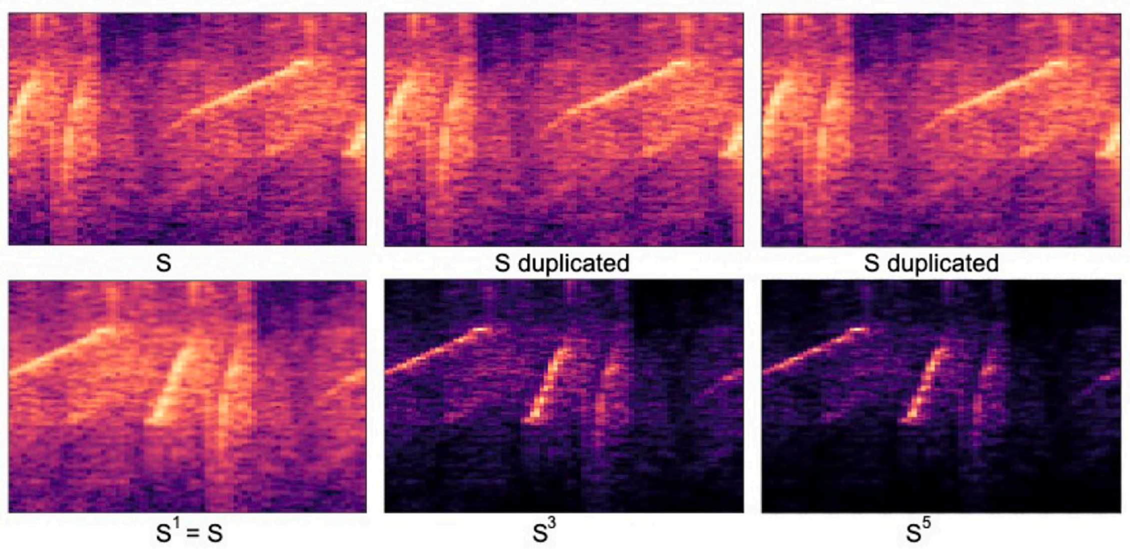


Fig. 2. Top: On the left is the original spectrogram (containing Hainan gibbon vocalisations), *S*, which is duplicated twice to form the 3 channel input requirement. In this study this input configuration is called *duplicate*. Bottom: On the left is the original spectrogram (containing Hainan gibbon vocalisations) for which two additional channels are formed by taking the exponent of 3 and 5. Since the spectrograms are normalised between [0, 1] the exponent makes the spectrogram darker. In this study this input configuration is called *exponent*.

Table 3
F1 score comparison of the two network input methods, duplication of spectrograms and applying an exponent to the spectrogram. The mean, minimum, maximum and standard deviation is provided for each dataset and input sample configuration. The results were obtained from 14 unique executions on each configuration. The feature extractor was frozen and thus only the weights in the classifier part of the CNN was optimised. For each dataset and input size configuration, the best result between exponent and duplication method is highlighted in bold. In all cases, the exponent method which achieved the best result compared to simply duplicating the spectrogram.

Dataset	Sample size	Input method	Mean	Min	Max	Sdev
Alethe	25	Exponent	**90.74**	83.50	**93.32**	2.36
Alethe	25	Duplicate	86.35	74.38	91.94	4.88
Alethe	50	Exponent	**93.56**	90.85	95.35	1.25
Alethe	50	Duplicate	89.18	80.56	95.35	1.25
Alethe	100	Exponent	**95.24**	91.65	**96.58**	1.27
Alethe	100	Duplicate	90.88	80.39	95.47	4.35
Lemurs	25	Exponent	**89.85**	87.74	**92.05**	1.35
Lemurs	25	Duplicate	88.52	85.10	91.35	1.99
Lemurs	50	Exponent	**93.39**	91.02	**95.37**	1.32
Lemurs	50	Duplicate	91.65	86.65	94.74	2.53
Lemurs	100	Exponent	**95.30**	91.42	**97.01**	1.39
Lemurs	100	Duplicate	93.59	86.95	96.42	2.87
Gibbons	25	Exponent	**94.89**	92.22	**97.26**	1.15
Gibbons	25	Duplicate	94.33	92.26	95.83	0.83
Gibbons	50	Exponent	**96.83**	95.42	**98.09**	0.68
Gibbons	50	Duplicate	96.52	95.45	97.44	0.54
Gibbons	100	Exponent	**97.96**	96.93	**98.74**	0.44
Gibbons	100	Duplicate	97.66	96.72	98.35	0.48

second representation to determine if classification performance could be improved by manipulating the spectrograms. We explored the effect of taking the values within the spectrogram and applying a constant exponent, say a, on each original spectrogram value to create a second channel, and a separate constant exponent, say b, on the same original spectrogram value to create the third channel. This second representation was denoted as *exponent*. In this spectrogram representation the first channel was the original spectrogram values (essentially an exponent of 1). Thus, if we denote the original spectrogram as S, then the three channels would be (S^1, S^a, S^b). We compared exponent values of (S^1, S^2, S^3) and then (S^1, S^3, S^5). Applying an exponent to a normalised spectrogram (for which the values are between 0 and 1) will result in the values becoming smaller. Thus, parts of the spectrogram with little sound will be decreased, while strong signals will still be visible. Fig. 2 illustrates this. The two representations were used as input to the 12 CNNs, the feature extractor was frozen (*no fine-tuning setting*) and we compared input sizes of 25, 50, 100 on three datasets (*lemurs*, *alethe* and *gibbons*). The results were collected over 15 unique executions.

The second experiment follows the findings from the first in that the best input representation was held constant, and then a comparison of the 12 CNNs was performed to determine the most suitable one. The feature extractor was frozen (*no fine-tuned setting*) and we compared input sizes of 25, 50, 100 on three datasets (*lemurs*, *alethe* and *gibbons*). The results were collected over 15 unique executions.

The third experiment also holds the input representation constant at the best value found in experiment 1, and then compares freezing or fine-tuning the feature extractor. This was done for the 12 CNNs to determine, firstly, which one would benefit the most from fine-tuning the feature extractor and secondly, to determine the relative performance when the feature extractor was frozen. Two configurations for the input size were explored (50 and 100 samples). These two were selected as it contains enough data to observe performance gains as a result of fine-tuning the feature extractor. For this experiment we ran 35 unique executions – a larger number of unique executions to provide a thorough investigation of fine-tuning for bioacoustics classification as performance gains were observed in computer vision research. Three datasets were used (*lemurs*, *alethe* and *gibbons*).

The fourth experiment holds the input representation and architecture constant at the best values found in experiment 1 and 2 respectively. This experiment was conducted to determine the performance on the four datasets when using 25, 50, 100 and 200 samples. Due to computational restrictions we only used three datasets in the first three experiments. For each dataset and input size configuration we compared the performance of the CNNs when the feature extractor was frozen and when fine-tuned. This was done to determine which approach is most suitable for bioacoustic problems for which there are data scarcity issues. The results were collected over 15 unique executions.

The software code was written in Python 3 for audio pre-processing and the general methodology, and the CNNs were implemented in Tensorflow 2 (Abadi et al., 2015). Each CNN was trained for 50 epochs (number of iterations of the CNN learning algorithm) using the Adam optimiser (Kingma and Ba, 2014) and a batch size of 32. The hyperparameters were obtained by conducting a random search using similar values to that in the study of Dufourq et al. (2021). Spectrograms were generated using the Librosa library (McFee et al., 2020).

4. Results

Under many conditions, CNNs pre-trained on the ImageNet dataset were able to produce classifiers which were able to identify calls in bioacoustic datasets with a high degree of accuracy (Table 3). Our comparison over the 12 pre-trained CNNs revealed that ResNet101V2 and ResNet152V2 produced the best results (Table 4). We compared the CNNs when the feature extractor was frozen and when it was fine-tuned and the difference in performance varied across the CNNs. Our findings reveal that when only 25 samples are used freezing the feature extractor results in CNNs that were as good as CNNs where the feature extractor was fine-tuned (Fig. 6). However, when more data was used, fine-tuning the feature extractor was the most optimal approach. Finally, we show that the performance of the CNNs can be improved when taking the

Table 4
Comparison of the average F1 score across the different network architectures and dataset configurations. The *exponent* approach was used for the spectrogram input. The feature extracted was frozen. The results are averaged across 13 unique executions. The results are ordered (highest to lowed) based on the average of each network architecture across all configurations. The best three performing network architectures on a particular dataset configuration is highlighted in bold.

Method	G 25	G 50	G 100	L 25	L 50	L 100	A 25	A 50	A 100
ResNet101V2	**95.30**	**97.40**	**96.27**	**92.05**	**94.92**	**97.01**	**92.10**	**95.37**	**98.36**
ResNet152V2	95.18	96.92	**96.58**	**91.42**	**95.31**	**96.62**	**93.32**	**94.94**	**98.35**
InceptionResNetV2	94.70	96.75	**96.57**	90.07	**95.35**	95.73	**92.73**	93.95	97.84
ResNet50V2	94.97	97.04	95.13	**91.96**	93.66	**96.36**	90.94	**94.82**	98.12
DenseNet169	94.92	96.95	95.69	89.33	93.78	95.59	91.76	93.32	97.95
DenseNet201	94.84	96.72	95.86	90.08	93.90	95.59	91.02	93.12	98.13
VGG16	**97.26**	**98.09**	94.99	87.74	92.93	95.01	90.26	92.32	**98.74**
DenseNet121	94.58	96.69	95.00	89.90	92.82	95.81	90.26	93.99	98.06
InceptionV3	92.22	95.42	95.40	88.72	93.23	95.45	91.21	93.28	96.93
ResNet101	**96.17**	**97.49**	94.23	90.01	92.21	91.42	91.00	91.02	97.80
Xception	93.88	95.79	95.50	88.15	93.81	94.10	90.74	91.12	97.51
MobileNetV2	94.62	96.65	91.65	88.78	90.85	94.91	83.50	93.40	97.71

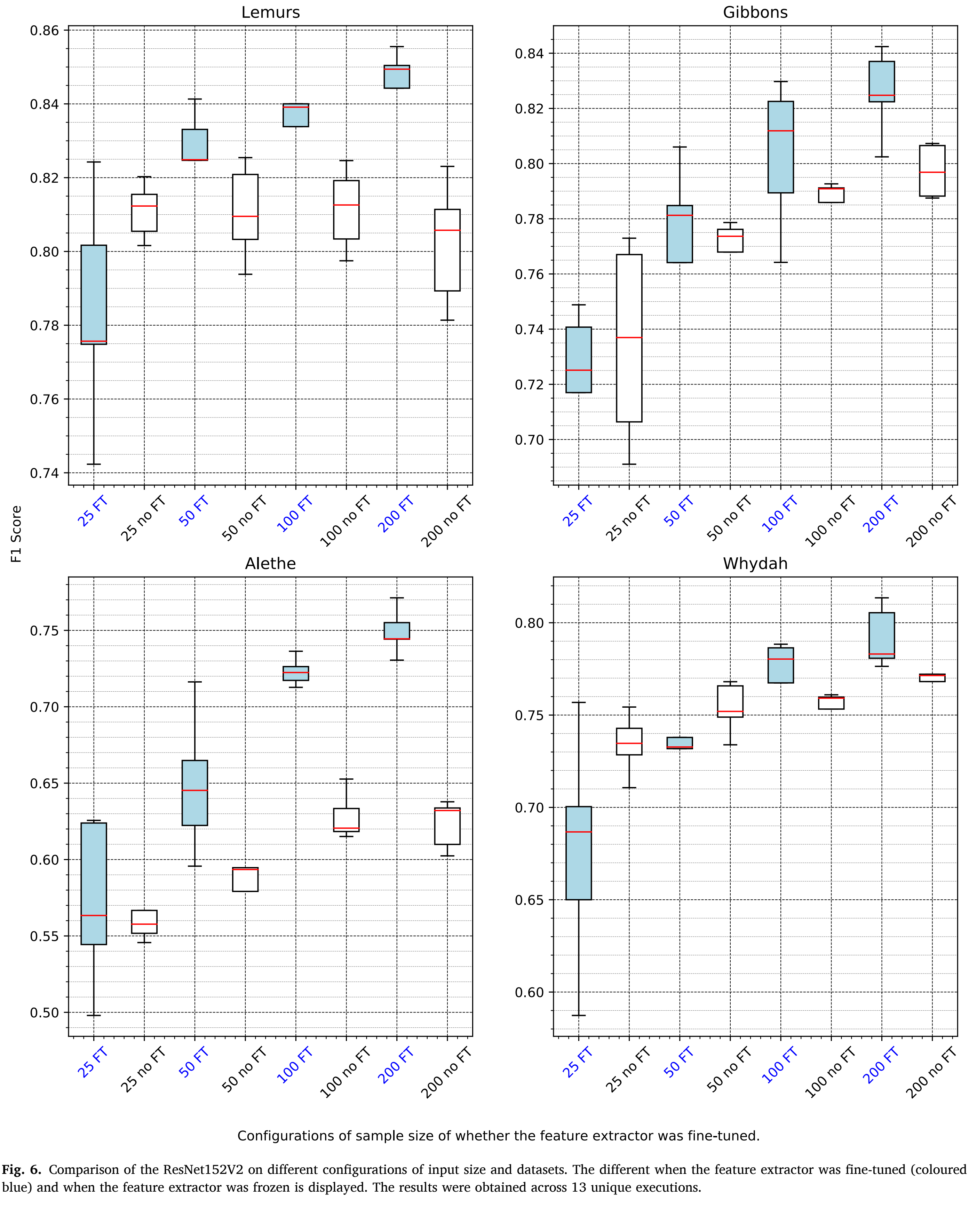


Fig. 6. Comparison of the ResNet152V2 on different configurations of input size and datasets. The different when the feature extractor was fine-tuned (coloured blue) and when the feature extractor was frozen is displayed. The results were obtained across 13 unique executions.

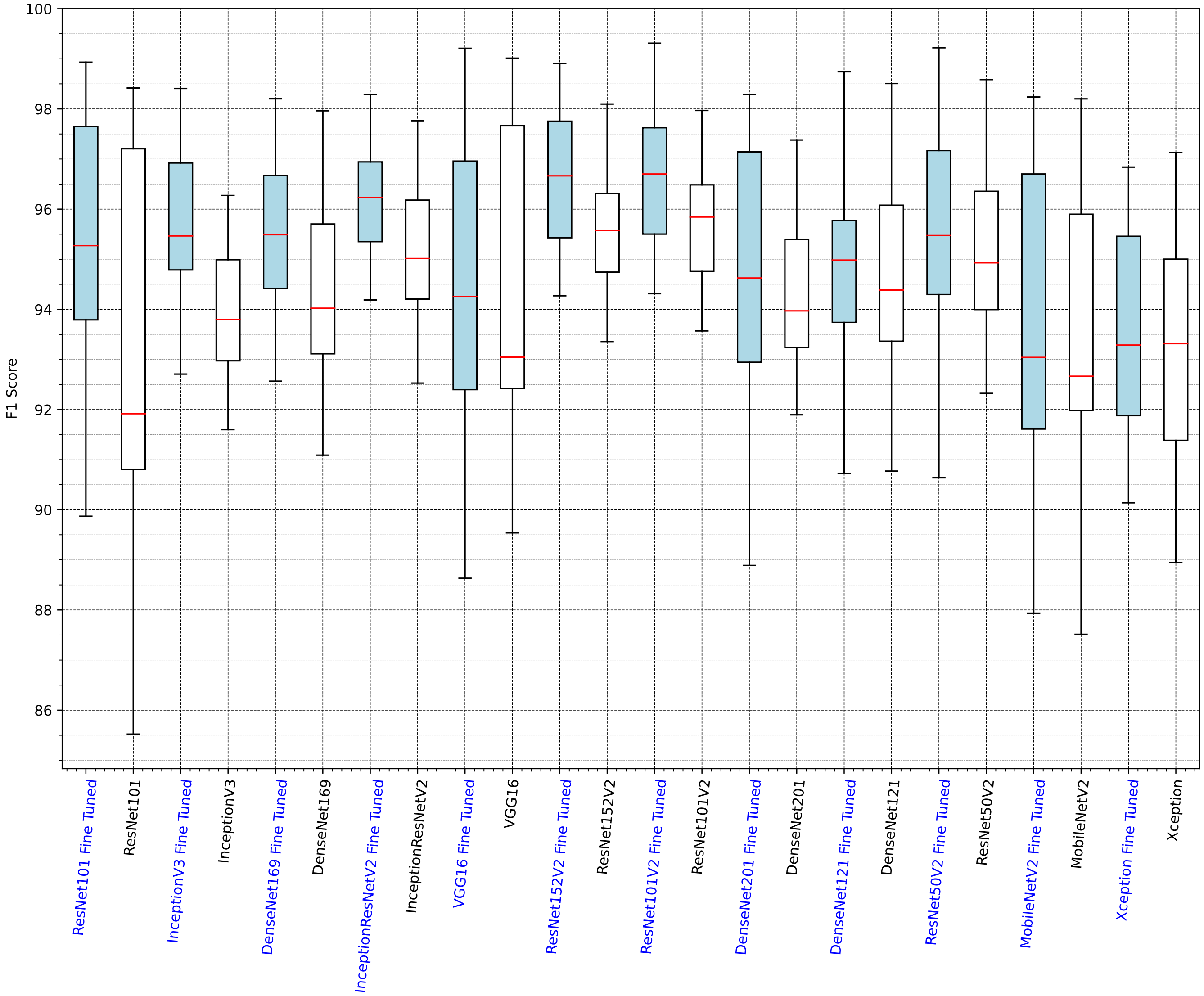


Fig. 3. F1 score comparison of the different architectures when using 50 samples for positive class across 37 unique executions of each architecture. Each architecture is displayed as a pair, with the feature extractor fine-tuned (coloured blue) and feature extractor frozen, and the results are ordered by difference in performance between fine-tuning and no fine-tuning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

exponent of the spectrograms as opposed to simply duplicating the spectrograms to create the 3 channel input (Table 3).

Table 3 presents the results for experiment 1. The findings reveal that the performance was on average better when the spectrograms were raised to a different exponent (*exponent*) as opposed to simply duplicating them (*duplicate*). This result was more apparent on the *alethe* dataset. On all the datasets and configurations the *exponent* approach obtained the highest F1 score, and the standard deviation was also on average smaller for the *exponent* approach. Following these findings, we used the *exponent* approach as input to the remaining experiments.

The findings for experiment 2 are presented in Table 4. The results are ordered based on the average performance across all of the datasets. ResNet101V2 resulted in the best overall performance (average of 95.42%) and closely followed by ResNet152V2 (average of 95.40%). MobileNetV2, on average, ranked last with an average performance of 92.45%. While ResNet101V2 ranked first on in this experiment, we selected ResNet152V2 for the remaining experiments due to the fact that ResNet152V2 outperforms ResNet101V2 (He et al., 2016b) on the ImageNet dataset.

Figs. 3 and 4 present the findings for experiment 3. The CNN results are presented in pairs (with fine-tuning of the feature extractor and without) and the results are ordered based on the difference of each pair. When 50 samples were used (Fig. 3), the greatest improvement in F1 score was obtained from ResNet101 (median improvement of 3.36%). An improvement was achieved by all CNNs except for Xception (median decrease of −0.03%). The best median performance was obtained by ResNet152V2 and ResNet101V2 for both the fine-tuning and no fine-tuning setting. The lowest standard deviation was obtained by ResNet152V2 for both fine-tuning (1.38) and no fine-tuning (1.19) suggesting that ResNet152V2 (no fine-tuning) was the most robust to the different training samples from each unique execution. On average across all CNNs, no-fine tuning resulted in a slightly lower standard deviation (2.07) compared to fine tuning (1.96). When 100 samples were used (Fig. 4), the greatest improvement was also obtained ResNet101 (median improvement of 1.51%). MobileNetV2 had a decrease in median performance of −1.33%. ResNet152V2 and ResNet101V2 obtained the best performance for both the fine-tuning and no fine-tuning setting. On average across all CNNs, no-fine tuning

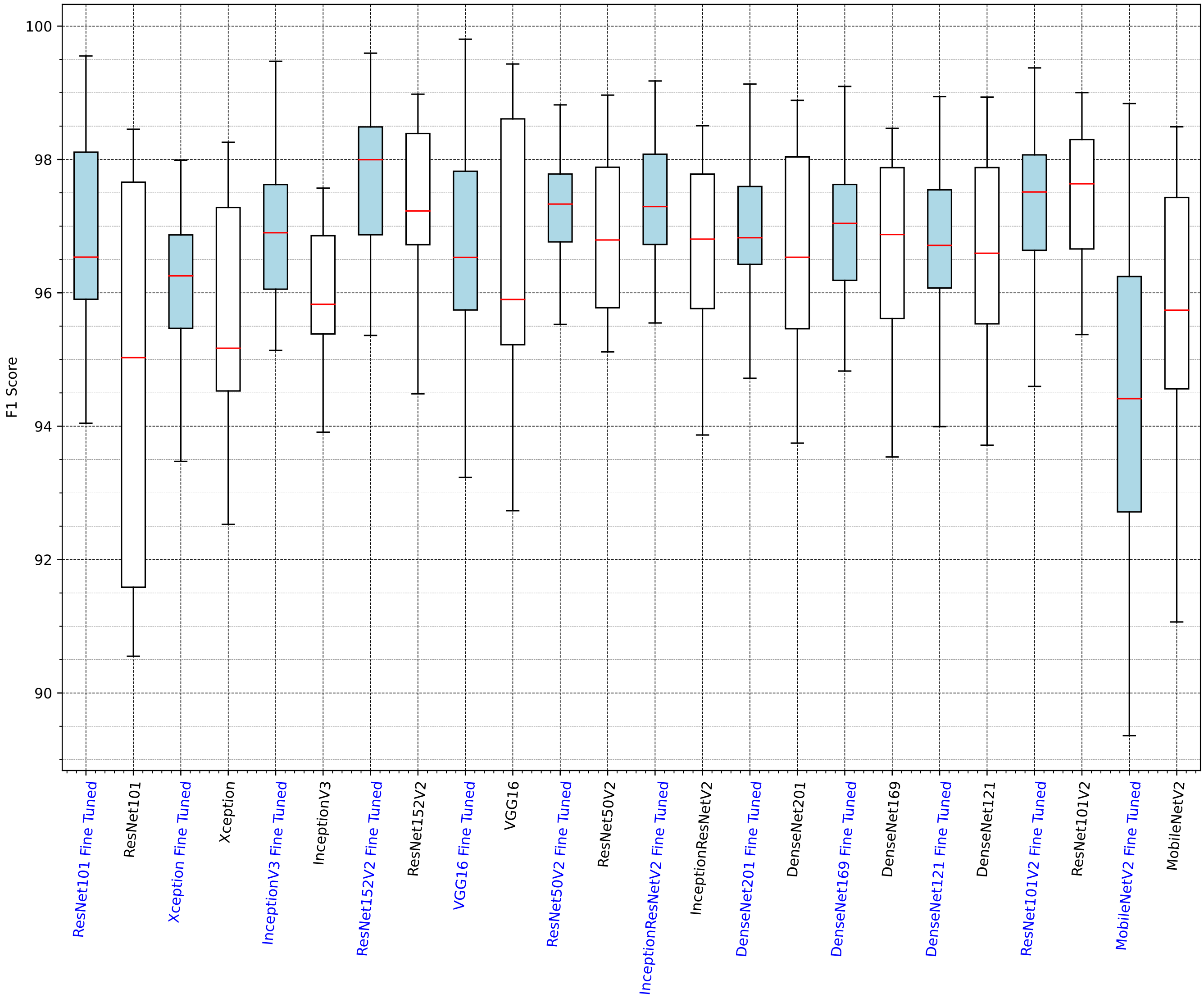


Fig. 4. F1 score comparison of the different architectures when using 100 samples for positive class across 37 unique executions of each architecture. Each architecture is displayed as a pair, with the feature extractor fine-tuned (coloured blue) and feature extractor frozen, and the results are ordered by difference in performance between fine-tuning and no fine-tuning.

resulted in a slightly higher standard deviation (1.61) compared to fine tuning (1.35).

The findings for experiment 4 are presented in Fig. 6. For an input of 25 samples, the performance when the feature extractor was not fine-tuned resulted in a higher median result on three datasets (*lemurs*, *gibbons* and *whydah*). For the *gibbons* dataset, the best result was achieved when 25 samples were used and the feature extractor was frozen compared to when the feature extractor was fine-tuned. For 25 samples, ResNet152V2 with no fine-tuning was robust to different training samples (smaller standard deviation on the *lemurs*, *alethe* and *whydah* datasets). These findings indicate that it is possible to use a pre-trained CNN and fine-tune the softmax output layer with as few as 25 samples and obtain good classifiers. This observation changed when the number of samples increased to 50, 100 and 200. When more input samples were used, the findings show that fine tuning the feature extractor resulted in better performance across all of the datasets.

5. Discussion

Continued advances in deep learning, computer vision and speech recognition offer many opportunities for bioacoustics research. While deep learning holds significant promise for the creation of PAM classifier models, CNNs would be even more widely used if they were easier to train, required less machine learning expertise to be implemented and if they could be trained on small datasets without overfitting. Transfer learning benefits from the fact that very little training is required and that fewer human decisions are required in the design of the architecture.

Using a pre-trained CNN feature extractor and adding a softmax output layer is less complex than having to optimise a CNN from scratch and requires less network design decisions and also less time on hyper-parameter tuning. We thus argue that this approach renders the use of deep learning much more accessible to practitioners. Extensive hyper-parameter tuning also requires expensive GPU hardware which might not be accessible to practitioners. Our findings revealed that results could be obtained on limited hardware within 9 h (10 epochs of fine-tuning the feature extractor) which would cost 2USD, at the time of writing, if that was executed on a Microsoft Azure virtual machine – thus rendering this approach affordable and accessible.

It is well accepted that no single machine learning algorithm – or in

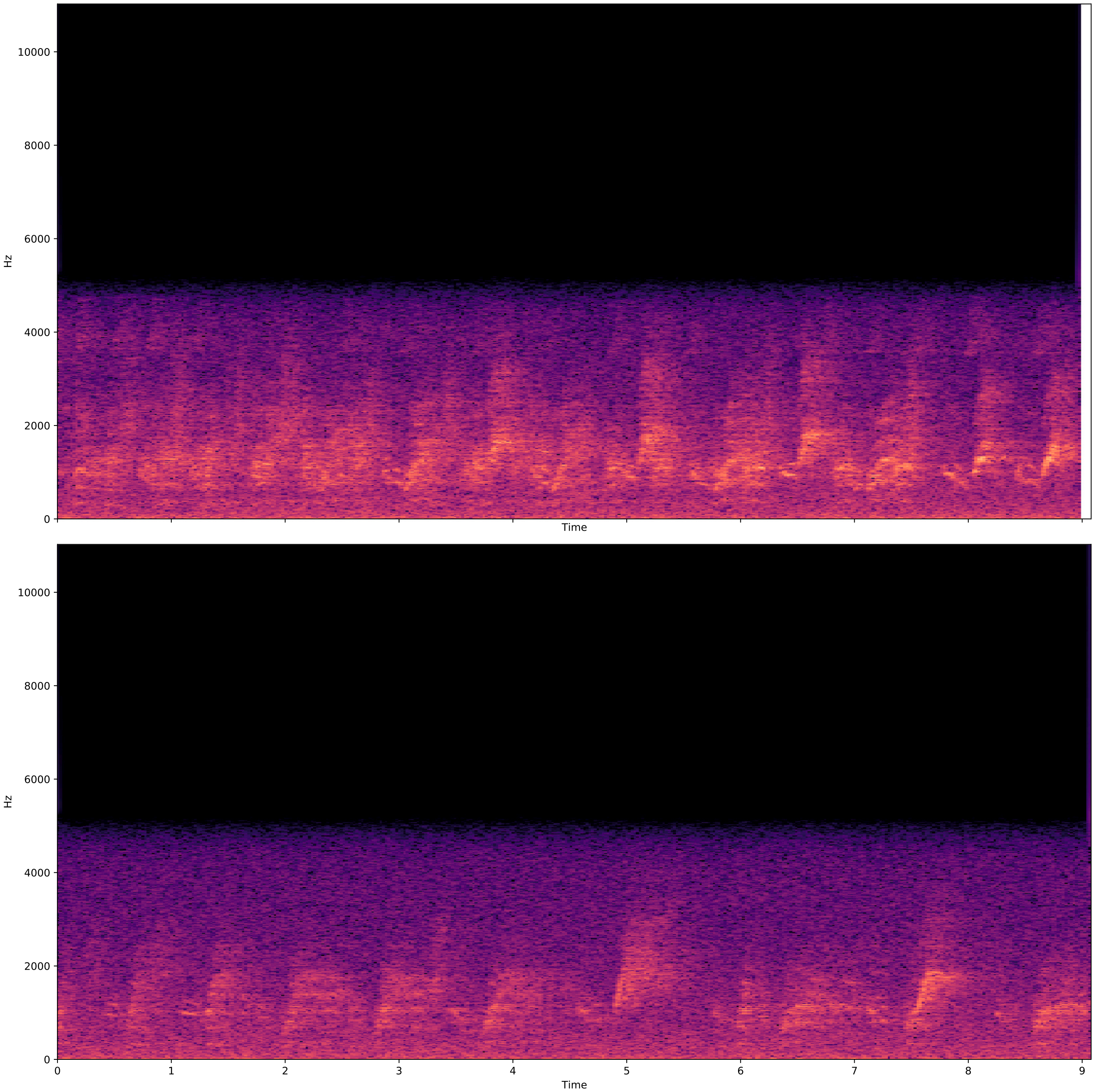


Fig. 5. Variation within calling bouts in the *lemurs* dataset. The spectrograms were pre-processed. Top: black-and-white ruffed lemur calls with more individuals calling simultaneously as compared to the bottom spectrogram.

this case a CNN – will consistently perform well across all applications and datasets. Table 4 shows that, VGG16 for example, obtained the best results on three configurations, but also did not achieve equally as good results on other configurations. ResNet101V2 and ResNet152V2 on average performed well across nearly all configurations and we thus recommend that either of these architectures are used as a starting point for researchers wanting to use pre-trained CNNs for bioacoustics research.

The experiments presented in Fig. 6 reveal that if very little data is available – in our case 25 examples – that pre-trained ResNet152V2 with the feature extractor frozen can yield good performance (up to 82% F1 score on the lemurs, 77% on the gibbons and 75% on the whydah dataset). This suggests that when conducting an acoustic survey, a practitioner can manually annotate a few examples and then start using a pre-trained ResNet152V2 model to find new examples. Once additional examples have been found via the pre-trained model, these new examples can be incorporated into the training set. This iterative process can be repeated until a large training set is obtained, after which, the pre-trained CNN can be fine-tuned to create a more robust classifier. We thus argue that practitioners can begin using CNNs relatively early on within a project to speed up the rate at which calls are found. These findings oppose existing knowledge that deep learning requires large training datasets. One possible explanation for the good performance achieved in this study is due to the high signal-to-noise ratio. It was also hypothesised that good performance was obtained due to the lack of variation within the calls (e.g. *gibbons* and *whydah* datasets), however this is not likely the case since the calls in the *lemurs* dataset have large variation within a calling bout – especially since a varying number of individuals call at the same time (Fig. 5).

CNNs are commonly executed on GPU hardware which results in faster training time. However, we deliberately trained ResNet152V2 on CPUs in an attempt to verify that training could be executed on less expensive hardware. We trained the CNN on a virtual machine running the “E2asV4” instance on Microsoft Azure with 16GB RAM and a 2.35Ghz AMD EPYC™ 7452 2 vCPU which at the time of writing cost 0.218 USD per hour. When the feature extractor was frozen, it took between 450 and 780 s to complete one epoch, and when the feature extractor was fine-tuned it took between 2035 and 3100 s per epoch. While these executions are time consuming, these findings reveal that it is possible to train pre-trained models on less expensive hardware making them accessible to researchers and practitioners.

Models pre-trained on ImageNet require a three channel input and it was unclear from the literature as to what is the best approach. A common approach is to simply duplicate the spectrogram. However, the findings in this study revealed that better performance can be achieved via manipulation of the spectrograms, in our case by taking exponents. This suggests that machine learning researchers should further explore this to determine if additional performance could be obtained by manipulating the input space. Similar findings have been observed in different areas of computer vision (Luo et al., 2017). In our investigation the exponents darkened areas of the spectrogam which had weaker signals and potentially could act as a means of reducing environmental noise to enable the CNN to learn better features for classification purposes. We also explored changing the fast Fourier transform window size between the 3 channels but this led to a decrease in performance.

6. Conclusion

This study is the first large-scale transfer learning experiment to compare a large number of modern CNNs across different bioacoustics datasets as a means to guide practitioners on how pre-trained CNNs can be used to facilitate and enable the creation of classification models. The emphasis of this study was to simplify the implementation as much as possible to demonstrate that less complexity is involved when using pre-trained models and that less annotated data is needed to train the models. This allows for more rapid development of classification models and less expert human time spent in manual annotation phases. Our findings reveal that reliable models can be trained with very little data (as a few as 25 calls). This will enable researchers to build models relatively early within the analysis phase of a project as only a few calls will need to be manually identified. We show that pre-trained models can be used in a low computational resource setting, thus enabling more researchers to implement this approach even if they do not have expensive GPUs. Our findings also reveal that performance can be optimised by manipulating spectrograms, which could be explored further. Additional experimentation on on dealing with very small datasets (<25 examples) would be beneficial to the research community. We hypothesis that Siamese neural networks would result in good model performanace and should be explored across various different species (Chicco, 2021; Acconcjaioco and Ntalampiras, 2021).

Credit author statement

ED and ID conceived the passive monitoring project and developed the study designs. ED conceived the development of the CNN implementations and designed the methodology. ED, CB and RF were responsible for fieldwork and data collection. ED and CB annotated the data. ED performed the analysis. ED and ID wrote the paper. All authors contributed critically to the drafts and gave final approval for publication.

ED: Conceptualization, Methdology, Software, formal analysis, writing - original draft

ID: conceptualization, writing - review & editing
CB: Data curation, writing - review & editing
RF: Data curation, writing - review & editing

Data accessibility

All code for training and testing the neural networks is available at https://github.com/emmanueldufourq/PAM_TransferLearning. A subset of acoustic recordings, including training and testing labels, has been stored on Zenodo and can be accessed for the *gibbons* (https://doi.org/10.5281/zenodo.6328319), *whydah* (https://doi.org/10.5281/zenodo.6330711), *lemurs* (https://doi.org/10.5281/zenodo.6331594) and *alethe* (https://doi.org/10.5281/zenodo.6328244).

Declaration of Competing Interest

The authors declare no competing interests.

Acknowledgements

ED is supported by a research chairship from the African Institute for Mathematical Sciences South Africa. This work was carried out with the aid of a grant from the International Development Research Centre, Ottawa, Canada, www.idrc.ca, and with financial support from the Government of Canada, provided through Global Affairs Canada (GAC), www.international.gc.ca. This work was supported by funding from Microsoft's AI for Earth program. We thank the management of Intaka Island Nature Reserve, South Africa for allowing us to conduct an acoustic survey. The audio data containing calls of the Thyolo Alethe was collected by an international team of researchers from the Biodiversity Inventory for Conservation (BINCO) and the Wildlife and Environmental Society of Malawi (WESM), led by RF (BINCO) and Tiwonge Gawa (WESM and the Malawi University of Science and Technology). The audio data containing calls of the black-and-white ruffed lemurs was collected by CB. CB acknowledges the support of the Centre ValBio research station & Madagascar National Parks for facilitating data collection.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems.

Acconcjaioco, M., Ntalampiras, S., 2021. One-shot learning for acoustic identification of bird species in non-stationary environments. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp. 755–762.

Anders, F., Kalan, A.K., Kühl, H.S., Fuchs, M., 2021. Compensating class imbalance for acoustic chimpanzee detection with convolutional recurrent neural networks. Ecol. Inf. 65, 101423.

Bergler, C., Schröter, H., Cheng, R.X., Barth, V., Weber, M., Nöth, E., Hofer, H., Maier, A., 2019. Orca-spot: An automatic killer whale sound detection toolkit using deep learning. Sci. Rep. 9, 1–17.

Bermant, P.C., Bronstein, M.M., Wood, R.J., Gero, S., Gruber, D.F., 2019. Deep machine learning techniques for the detection and classification of sperm whale bioacoustics. Sci. Rep. 9, 1–10.

Chicco, D., 2021. Siamese neural networks: an overview. Artif. Neural Netw. 73–94.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.

Çoban, E.B., Pir, D., So, R., Mandel, M.I., 2020. Transfer learning from youtube soundtracks to tag arctic ecoacoustic recordings. In: ICASSP 2020-2020 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP). IEEE, pp. 726–730.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition. IEEE, pp. 248–255.

Disabato, S., Canonaco, G., Flikkema, P.G., Roveri, M., Alippi, C., 2021. Birdsong detection at the edge with deep learning. In: IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, pp. 9–16.

Dufourq, E., Durbach, I., Hansford, J.P., Hoepfner, A., Ma, H., Bryant, J.V., Stender, C.S., Li, W., Liu, Z., Chen, Q., Zhou, Z., Turvey, S.T., 2021. Automated detection of hainan gibbon calls for passive acoustic monitoring. Remote Sens. Ecol. Conserv. 7, 475–487.

Efremova, D.B., Sankupellay, M., Konovalov, D.A., 2019. Data-efficient classification of birdcall through convolutional neural networks transfer learning. In: Digital Image Computing: Techniques and Applications (DICTA). IEEE, pp. 1–8.

Escobar-Amado, C.D., Badiey, M., Pecknold, S., 2022. Automatic detection and classification of bearded seal vocalizations in the northeastern chukchi sea using convolutional neural networks. J. Acoust. Soc. Am. 151, 299–309.

Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M., 2017. Audio set: An ontology and human-labeled dataset for audio events. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP). IEEE, pp. 776–780.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp. 249–256.

Hawkins, D.M., 2004. The problem of overfitting. J. Chem. Inf. Comput. Sci. 44, 1–12.

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

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks. In: European conference on computer vision. Springer, pp. 630–645.

Henri, E.J., Mungloo-Dilmohamud, Z., 2021. A deep transfer learning model for the identification of bird songs: A case study for mauritius. In: International Conference on Electrical, Computer Communications and Mechatronics Engineering (ICECCME). IEEE, pp. 01–06.

Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al., 2017. Cnn architectures for large-scale audio classification. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp. 131–135.

Hill, A.P., Prince, P., Snaddon, J.L., Doncaster, C.P., Rogers, A., 2019. Audiomoth: a low-cost acoustic device for monitoring biodiversity and the environment. HardwareX 6, e00073.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications arXiv:1704.04861.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.

Incze, A., Jancsó, H.B., Szilágyi, Z., Farkas, A., Sulyok, C., 2018. Bird sound recognition using a convolutional neural network. In: IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 000295-000300.

Jaramillo, J.C.A., Murillo-Fuentes, J.J., Olmos, P.M., 2018. Boosting handwriting text recognition in small databases with transfer learning. In: 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, pp. 429–434.

Jiang, J.j., Bu, L.r., Duan, F.j., Wang, X.q., Liu, W., Sun, Z.b., Li, C.y., 2019. Whistle detection and classification for whales based on convolutional neural networks. Appl. Acoust. 150, 169–178.

Khalighifar, A., Brown, R.M., Vallejos, J.G., Peterson, A.T., 2021. Deep learning improves acoustic biodiversity monitoring and new candidate forest frog species identification (genus Platymantis) in the philippines. Biodivers. Conserv. 30, 643–657.

Khalighifar, A., Jiménez-Garc&rsquo;í, D., Campbell, L.P., Ahadji-Dabla, K.M., Aboagye-Antwi, F., Ibarra-Juárez, L.A., Peterson, A.T., 2022. Application of deep learning to community-science-based mosquito monitoring and detection of novel species. J. Med. Entomol. 59, 355–362.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization https://arxiv.org/abs/1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Kübra Karaca, B., Oltu, B., Özgür, A., Erdem, H., 2021. Comparison of transfer learning strategies for diabetic retinopathy detection. In: Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–5.

LeBien, J., Zhong, M., Campos-Cerqueira, M., Velev, J.P., Dodhia, R., Ferres, J.L., Aide, T. M., 2020. A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network. Ecol. Inf. 59, 101113.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1989. Handwritten digit recognition with a back-propagation network. Adv. Neural Inf. Process. Syst. 2.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324.

Leroux, M., Al-Khudhairy, O.G., Perony, N., Townsend, S.W., 2021. Chimpanzee voice prints?. insights from transfer learning experiments from human voices, p. 08165 arXiv:2112.

Liu, Y., Yang, C., Liu, K., Chen, B., Yao, Y., 2019. Domain adaptation transfer learning soft sensor for product quality prediction. Chemometr. Intell. Lab. Syst. 192, 103813.

Lopez, A.R., Giro-i Nieto, X., Burdick, J., Marques, O., 2017. Skin lesion classification from dermoscopic images using deep learning techniques. In: 13th IASTED international conference on biomedical engineering (BioMed). IEEE, pp. 49–54.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G., 2015. Transfer learning using computational intelligence: a survey. Knowl. Based Syst. 80, 14–23.

Lu, T., Han, B., Yu, F., 2021. Detection and classification of marine mammal sounds using alexnet with transfer learning. Ecol. Inf. 62, 101277.

Luo, Z., Chen, J., Takiguchi, T., Ariki, Y., 2017. Facial expression recognition with deep age. In: IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, pp. 657–662.

McFee, B., Lostanlen, V., McVicar, M., Metsai, A., Balke, S., Thome, C., Raffel, C., Malek, A., Lee, D., Zalkow, F., Lee, K., Nieto, O., Mason, J., Ellis, D., Yamamoto, R., Seyfarth, S., Battenberg, E., Bittner, R., Choi, K., Moore, J., Wei, Z., Hidaka, S., nullmightybofo, Friesch, P., St&rdquo;oter, F.R., Hereñú, D., Kim, T., Vollrath, M., Weiss, A., 2020. Librosa.

Mehra, R., et al., 2018. Breast cancer histology images classification: Training from scratch or transfer learning? ICT Express 4, 247–254.

Morgan, M., Braasch, J., 2021. Long-term deep learning-facilitated environmental acoustic monitoring in the capital region of new york state. Ecol. Inf. 61, 101242.

Nanni, L., Maguolo, G., Paci, M., 2020. Data augmentation approaches for improving animal audio classification. Ecol. Inf. 57, 101084.

Nolasco, I., Terenzi, A., Cecchi, S., Orcioni, S., Bear, H.L., Benetos, E., 2019. Audio-based identification of beehive states. In: ICASSP 2019-2019 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP). IEEE, pp. 8256–8260.

Ntalampiras, S., Kosmin, D., Sanchez, J., 2021. Acoustic classification of individual cat vocalizations in evolving environments. In: 44th International Conference on Telecommunications and Signal Processing (TSP). IEEE, pp. 254–258.

Pamula, H., Pocha, A., Klaczynski, M., 2020. Deep learning methods for acoustic monitoring of birds migrating at night. In: Forum Acusticum, pp. 2761–2764.

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359.

Paumen, Y., Mälzer, M., Alipek, S., Moll, J., Lüdtke, B., Schauer-Weisshahn, H., 2021. Development and test of a bat calls detection and classification method based on convolutional neural networks. Bioacoustics 1–12.

Ruff, Z.J., Lesmeister, D.B., Duchac, L.S., Padmaraju, B.K., Sullivan, C.M., 2020. Automated identification of avian vocalizations with deep convolutional neural networks. Remote Sens. Ecol. Conserv. 6, 79–92.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.

Sankupellay, M., Konovalov, D., 2018. Bird call recognition using deep convolutional neural network, resnet-50. In: Proc. ACOUSTICS, pp. 1–8.

Shao, S., McAleer, S., Yan, R., Baldi, P., 2018. Highly accurate machine fault diagnosis using deep transfer learning. IEEE Trans. Ind. Inf. 15, 2446–2455.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition arXiv:1409.1556.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence.

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.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A survey on deep transfer learning. In: International conference on artificial neural networks. Springer, pp. 270–279.

Thakur, A., Thapar, D., Rajan, P., Nigam, A., 2019. Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss. J. Acous. Soc. Am. 146, 534–547.

Waddell, E.E., Rasmussen, J.H., Širović, A., 2021. Applying artificial intelligence methods to detect and classify fish calls from the Northern Gulf of Mexico. J. Mar. Sci. Eng. 9, 1128.

Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. J. Big Data 3, 1–40.

Xie, J.j., Ding, C.q., Li, W.b., Cai, C.h., 2018. Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks arXiv preprint arXiv:1803.

Yi, Z., Wang, Y., 2021. Transfer learning on interstitial lung disease classification. In: International Conference on Signal Processing and Machine Learning (CONF-SPML), pp. 199–205.

Zhong, M., LeBien, J., Campos-Cerqueira, M., Dodhia, R., Ferres, J.L., Velev, J.P., Aide, T. M., 2020. Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling. Appl. Acoust. 166, 107375.

Zhong, M., Taylor, R., Bates, N., Christey, D., Basnet, H., Flippin, J., Palkovitz, S., Dodhia, R., Ferres, J.L., 2021. Acoustic detection of regionally rare bird species through deep convolutional neural networks. Ecol. Inf. 101333.