Versatile ecoacoustic AI recognisers created with 1-D CNNs and varied sampling strategies.

Peter Griffioen(1), Lindy Lumsden(1), Louise Durkin(1), and Lachlan Francis(1)

(1) Arthur Rylah Institute for Environmental Research, Department of Energy, Environment and Climate Action, Heidelberg, Victoria 3084, Australia

ABSTRACT

To efficiently analyse large acoustic datasets collected for birds, frogs, bats and terrestrial mammals, we developed a 1-dimensional Convolutional Neural Network (CNN) system of models which provides advantages over more commonly employed 2-D networks. Like a 2-D 'image-recognition' network, the 1-D network may accept spectrographic representation of the audio, but additionally it can utilise any other time-dependent indices of audio information. A further advantage is that 1-D CNNs are not required to 'complete-the-square' image that is commonly required to shoehorn the audio dataset to 2-D CNNs. Thus 1-D CNNs can accept a wider variety of information and better match the structure of the data. This becomes apparent when considering the diversity of ecoacoustics applications supported by our 1-D CNN system. Microbat research requires audio at 192-500 kHz sampling rates. Frog, bird and koala datasets can use lower sampling rates and our library for these groups comprises recordings of 48 kHz, 44.1 kHz, 24 kHz or even 22.05 kHz. Models must be tuned not only to their target species, but also to their target dataset and thus are generally bespoke. A 1-D CNN system, combined with custom data sampling strategies and a database to keep track of the design, production and application of the models, allows the efficient production of bespoke models with high accuracy classification. These models have been used to process acoustic data on a single workstation at rates up to 300 seconds/second. This effectively means that one year's worth of 24/7 recordings can be processed in a little over a day on a moderately powerful workstation. We present two models exemplifying the system's utility and accuracy. The first is a bat call recogniser model for 16 species in southwest Victoria, Australia. This model accepts recordings at 192 kHz and above to process 0.75 second sound samples. It supplies these samples as 1,124 frames x 163 parameters matrices to a 1-D CNN created within TensorFlow. It has an average accuracy of 90.5% for species identification. The second model is for 15 frog species in Victoria, for application to recordings at 48 kHz. This model is based on 70 frames of 163 parameters per 1.5-second sound sample and averages 96.8% accuracy in species identifications. All models created within our system are supported by our field data processing software ARISA and validation using our publicly available software ARIEL.

1 Introduction

The availability of reliable battery-operated audio recorders has provided substantial opportunities to ecologists for efficient monitoring of fauna via their sounds. Over the past eight years, we have been employing passive acoustic monitoring (PAM) for a variety of fauna survey applications across Victoria, Australia. The cumulative PAM effort has amassed a substantial acoustic dataset, with approximately 100 TB of bat and 100 TB on non-bat data (targeting frogs, birds, koalas and other vocalising vertebrates). The non-bat recordings are equivalent to 20 years of continuous recording. Datasets of this magnitude need automated methods to analyse them that are adaptive, efficient and accurate. We here describe our design and workflow, and the system developed to enable this rich acoustic dataset to be investigated.

1.1 Related Works

Using convolutional neural networks (CNNs) for the purpose of sound classification and species identification has resulted in many successes. Early works such as Piczak (2016), used three relatively shallow CNNs to classify 999 bird species with a moderate average precision of 41.7%. More recently Ruff et al. (2019), used spectrograms of several North American owl species to train CNNs to identify calls within recordings at 63-91% accuracy depending upon the species. Similarly, Karl et al. (2021) achieved a mean average classification precision of 79.1% for North American bird species with BirdNET. Nanni et al. (2019) used ensembles of CNNs to

Winner of President's Medal for Best Paper at Acoustics 2025

identify calls of birds, bats and whales with varying success. Mel-spectrograms were used to train a CNN to identify 24 species of birds and frogs with a mean-average-precision of 89.3% (LeBien et al., 2020). The variety of species sounds and calls that can and have been investigated with CNNs is perhaps matched by the variety of the designs of classifiers using CNNs (Kritchen 2023).

Many deep-learning species identification models employ 2-dimensional convolutional neural network (2-D CNN) design, generally associated with image recognition, to analyse spectrograms. This process mimics human recognition of characteristic spectrograms by researchers. Pre-trained image classifiers are readily available which can be tuned to accept images of spectrograms augmented by custom output layers to identify species of interest (Caravalho 2021, Elchinski et al., 2022, Himawan et al., 2018, LeBien et al., 2020, Nanni 2020). This provides highly optimised and well-trained networks which can be quickly implemented with very good results (LeBien et al., 2020). However, the use of a pre-trained model for the bulk of the classification calculation requires that the data presented to the model is in the exact format of its original design. In the case of the references above this has meant rescaling the image of the spectrogram to a square 224x224, 227x227 or 299x299 pixel images (Nanni et al., 2020).

Custom 2-D audio classifiers are not as restricted to providing a square image. Ruff et al. (2019) utilise a 500x129 input matrix, Pizack (2016) use a 170x430 matrix, Kahl et al. (2021) use 384x64, and Xie et al. (2022) implemented multiple sizes. In addition to requiring custom software, these CNNs needed to be entirely trained with the labelled species data rather than having just the output layers trained. This increases both the computing and data resources required for working models.

All 2-D CNNs are built up by stacking layers of kernels strided horizontally and vertically across the image, often 3x3 pixels at a time (Kritchen 2023). Optimised for photographs, this process aids the detection of characteristic components of the image such as a shape or image density change but loosens the relationship of where it is in the image both horizontally and vertically. However for spectrograms, the X axis denotes time and the Y axis frequency. Striding the frequency axis may result in similar-shaped patterns of spectrograms being confused as the frequency scale is untethered. This may result in false positives from sounds with similar spectrographic shapes. Spectral filtering may counter this but can limit the variety of species calls that can be detected (Xie et al., 2022).

Less common are the use of 1-dimensional convolutional neural networks (1-D CNNs) for audio analysis. A 1-D CNN differs to a 2-D network as it constrains the stride window to the X axis or time domain. 1-D design strategies may use raw audio streams for classifying sounds (Abdoli 2019, Abdullah et al., 2022) or classifying music into genres (Allay & Koerich 2021). The use of spectrographic information in a 1-D CNN is another approach (Sharan et al., 2021). The frequency bands are added as channels identically to red-green-blue (RGB) in a 2-D network. The channels are fully connected in the first layer. Combinations maintain their influence similar to red and green combining to make yellow in the visual spectrum, and these combinations may be made into features. In effect, the features are now combinations of lines (1-D) rather than shapes (2-D). Sharon et al. (2021) explores the input of spectrogram derivatives such as the smoothed spectrograms, Mel-spectrograms and cochleagrams and the combination of these signal representations into ‘fusion’ networks. It is these ‘fusion’ networks that produced the best results, and this is the type of network present in our system.

2 METHODS

For the design of our audio processing and species identification system, we considered the following aspects.

- The Arthur Rylah Institute (ARI) maintains a library of field sound recordings that can be interrogated to supply training data for the audio recognisers. These data will be re-analysed with future models to detect species previously missed or not studied. Besides target species’ calls, models are trained with non-target sounds, hereafter referred to as ‘Noise’, that are likely to be encountered in field recordings. Additionally non-bat models were supplied noise samples of novel sounds which may aid model development (e.g. music).
- We considered the bats separately from other terrestrial fauna, and focus just on the smaller, insectivorous, echolocating species (hereafter called ‘bats’), not flying-foxes and fruit bats. The calls of most species of echolocating bats are high frequency and so need to be recorded at much higher sampling rates than the other fauna. Therefore, separate but similar model designs are used for the bats compared to other fauna groups. For bats, a 0.75 second audio sample, recorded at a sample rate of 192 kHz, appeared to provide sufficient information to differentiate between species to the level that experts can discern these species from spectrograms. All of ARI’s bat recordings are at either 192 kHz or 384 kHz making 192 kHz a natural choice.

- Similarly for other fauna, a 1.5 second audio sample of good quality was deemed sufficient to identify the vast majority of frog, bird and non-flying mammal (hereafter 'mammal') calls. Longer calls, such as that of the Laughing Kookaburra (*Dacelo novaeguineae*) are often repetitive or have sufficient audio information within the 1.5 second sample, such that they can be identified by a human observer. In addition, isolating single species calls is increasingly becoming difficult with longer sampling periods for many species, due to the presence of other species and unwanted noises. Most non-bat audio recordings collected by ARI are at a 48 kHz sample rate and this rate is used as the default for audio processing.
- Audio datasets collected by research partners and clients are often collected at other sampling rates such as 44.1 kHz, 24 kHz and 22.05 kHz. In some cases, these datasets constitute the majority of target species training calls available. Our audio recognisers support both the incorporation these data for training and the processing of these datasets.

We developed a 'fusion' type 1-D CNN framework for our models which makes use of both spectrograms and many other time-related parameters derived from encountered sounds. 1-D CNNs can better cope with varying sample lengths encountered in audio (Sharan et al., 2021), and this is a practical consideration given the varied sampling rates of the audio collected for studies these models are applied to. We started with two base audio sampling strategies, one for high sample rate recordings of bats and the other for other fauna such as frogs, birds and mammals. These base sampling strategies were then modified into custom sampling strategies either for specific species or the model's application to large existing datasets.

2.1 Base bat sampling strategy

A 192 kHz sampling rate was selected as the model standard as this is the minimum frequency commonly used in quality field recorders. Also, the fundamental frequencies of the calls of all target species from southwest Victoria were encapsulated with the 96 kHz audio frequency range this offers, with only a small proportion of high harmonics of some species outside this range. Input data were converted from stereophonic to monophonic form. A 0.75 second exemplar sample at 192 kHz provides 144,000 measurements per sample for analysis. Each of the exemplars were partitioned into 281 non-overlapping frames of 512 measurements for analysis, each frame being a 2.6 millisecond subsample. A decibel spectrogram for each of the 281 frames was calculated in the software Librosa (McFee et al., 2015) using a short-term Fourier transform. The 256 (+1 zero frequency) frequencies were subsampled to 128 (+1 zero frequency) frequencies and normalised to a -1 to +1 range. The result was a 281x129 matrix representing the spectral data. In addition, the 281 frames of sound data were analysed with PyAudioAnalysis (Giannakopoulos 2015) to extract summary short-term features for each of the 281 frames. The high-frequency bat data were analysed as if they were within human auditory range as the PyAudioAnalysis was intentionally misinformed that the data was recorded at 48 kHz. This has the effect of stretching the 0.75 second sample to a 3 second sample. PyAudioAnalysis produces 34 features such as energy, spectral centroid, Mel Frequency Cepstral Coefficients (MFCCs) and Chroma Vectors. A full list of all parameters is provided in Table 1. These data were also normalised by the ranges observed across thousands of samples. These 281x34 features were then appended to the spectral data matrix resulting in 281x163 exemplar matrices that constituted the raw data for the neural network.

Table 1. The 34 short-term features provided by PyAudioAnalysis used in addition to the dB spectrogram to characterise the sounds (Giannakopoulos 2015).

Feature ID	Feature	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal for a frame.
2	Energy	The sum of squares of the signal values.
3	Entropy of Energy	The entropy of sub-frames' normalized energies.
4	Spectral Centroid	The centre of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients where the frequency bands are distributed according to the mel-scale.
22-33	Chroma Vector	Spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

2.2 Base frog, bird and mammal strategy

Our non-bat recordings use a sample rate of 48 kHz. A 1.4933 second mono audio sample is used in the base strategy. This provides 71,680 measurements per sample, which are partitioned into 70 non-overlapping frames of 1,024 measurements for analysis, each frame being a 21.3 millisecond subsample. Similar to the bats, a decibel spectrogram for each of the 70 frames was calculated in the software Librosa using a short-term Fourier transform, with the 512 (+1) frequency bins subsampled to 128 (+1 zero frequency) frequencies. The 70 frames were analysed with PyAudioAnalysis at 48 kHz producing the same 34 features per frame. This results in a 70x163 matrix of -1 to +1 normalised values that are provided to the neural network (Figure 1).

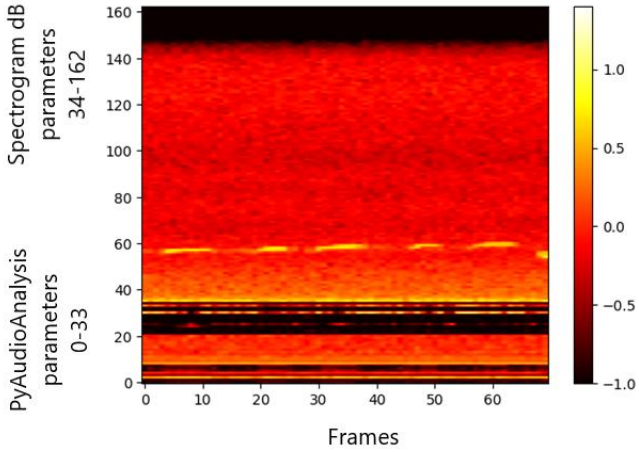

Figure 1. An example (using a Ground Parrot *Pezoporus wallicus* call) of a data matrix call normalised to a -1 to +1 scale. The PyAudioAnalysis parameters are in rows 0 to 33, and the spectrogram frequencies are in rows 34 to 162 for 70 frames.

2.3 Varied sampling strategies

By implementing alternative sampling strategies (Table 2), we were able to streamline the production and application of the audio recogniser models. For model production, the sampling strategies aid the standardisation of the frames and parameters, the defining matrix dimensions, expected by the model. This allows the re-use of the same CNN designs by all strategies that have the same set of frames and parameters. In situations where the frames are a multiple of 2 or 4 of that of the base strategy (see PreciseBat and BirdFrogSlowWide below), the judicious addition of simple MaxPooling layers or tweaking of kernel and stride parameters in the CNN (Kritchen 2023) would allow the reuse of a base design.

Models made with a chosen sampling strategy must be applied to field data processed with the same sampling strategy. By storing a JSON file describing the sampling strategy with the model, generic model application software was then developed and informed as to how to preprocess the audio data for application in that model.

The sampling strategies can also be used to coalesce training data recorded at different sampling rates. For example, the large datasets collected to detect Koalas (*Phascolarctos cinereus*) in Victoria, have been recorded at lower sampling rates (24 kHz) rather than the 48 kHz of frog and bird surveys. As a result, almost all of the Koala call training data is derived from these 24 kHz files. Models made with the ‘24kHzRestrict’ strategy can be applied without resampling to 48 kHz field files, using both 24 kHz Koala training data and 48 kHz training noise data. This is because the strategy restricts the spectrograms to 12 kHz audio which is available to all the training data. This restriction is reproduced when processing the field files. Similarly, for Koala datasets from New South Wales (NSW) recorded at 22.05 kHz, we use a custom sampling strategy which resamples all training data to this sampling rate and adjusts the sample step between frames such that the data fits the 70x163 template. This creates a custom model tuned to NSW field files, of which there are many (>100TB).

Table 2. The sampling strategies used to tune models for their training and application datasets. Each strategy is defined by custom input, processing sampling rates, sample durations and analysis frame sizes and steps. The first five are bat sampling strategies with the remainder being the other groups. SR: sampling rate; FFT: fast Fourier transform.

Strategy Name	Description	SR	PyAudio SR	Duration (sec)	Frames / Parameters	Sample Length	Sample Width / Step	Audio Freq Mask
DefaultBat	PyAudio set to 48000	192000	48000	0.74933	281/163	143872	512/512	
SlowBat	Slow PyAudio	192000	12000	0.74933	281/163	143872	512/512	
SlowFilteredBat	Slow PyAudio Filtered	192000	12000	0.74933	281/163	143872	512/512	>7.5kHz
PreciseBat	High sampling bats	192000	12000	0.75000	1124/163	144000	256/128	
PreciseBatFiltered	High sampling with filter	192000	12000	0.75000	1124/163	144000	256/128	>7.5kHz
DefaultFrogBird	Frog/bird with 129 FFT bins	48000	48000	1.49333	70/163	71680	1024/1024	
BirdFrogSlowAudio	Slow audio down	48000	12000	1.49333	70/163	71680	1024/1024	
GroundParrotSlow	Filter Ground Parrot	48000	12000	1.49333	70/163	71680	1024/1024	2.4375-6.5625kHz
BirdFrogSlowWide	High sampling birds	48000	12000	1.49333	140/163	71680	512/512	
24kHzRestrict	Filter to 24kHz and below	48000	12000	1.49333	70/163	71680	1024/1024	<12kHz
KoalaAudio24kHz	Slow audio down	24000	12000	1.49333	70/163	35840	512/512	
NSWKoala	22.050kHz recordings	22050	22050	1.48608	70/163	32768	512/466	

Finally, fine tuning of the data to aid detection of the target species can be implemented by tuning the sampling strategies. For example, the PyAudioAnalysis parameters are dominated by the 12 Mel Frequency Cepstral Coefficients (MFCC) which are tuned to the human voice. By dropping the audio frequencies, which are up to 24 kHz for the 48 kHz sampling rate files, into the range of frequencies of the human voice (<8 kHz), the information content of the MFCCs is increased. This results in improved accuracy of model fits and was achieved by simply setting the sample rate to 12 kHz within the PyAudioAnalysis short-term features function (Figure 2). Additionally, filtering can be applied directly to the spectrogram component of the input matrix as required for the species being detected (Figure 2 right).

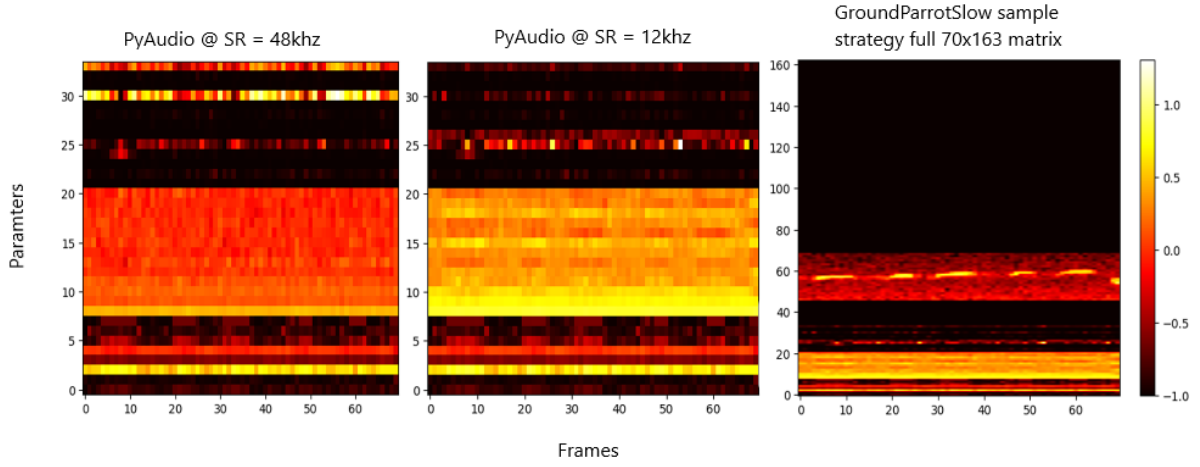

Figure 2. Adjusting the sample rate for the 34 PyAudioAnalysis short-term features from observed SR @ 48 kHz (left) to MFCCs range 12 kHz (middle). Upper and lower spectrogram masking evident in the GroundParrotSlow sample strategy (right) once the full 70x163 matrix is formed.

2.4 Network designs

We currently have 12 bat models and 16 bird/frog/mammal models which can be matched with their corresponding sampling strategies to create models. The sample bat and frog models presented are described in

general CNN terms as defined in Kritchen (2023). For each training exemplar, model data consists of the output matrix as defined by the sampling strategy combined with a '1-hot-vector' label data. This has a 1 in the appropriate label class column and 0 in all other class columns. All convolutional layers within both networks use Relu activation except for the output layer which uses a Softmax function to provide probability-like estimates across the output classes.

The bat model 1-D CNN is a relatively simple design of 7 convolutional layers and a dense output layer, all of which contain approximately 540,000 trainable parameters. The frog model CNN is an example of a deep ResNet design (He et al., 2015) and contains 30 convolutional layers, a dense output layer and consists of 2.76 million parameters. It's Identity Resnet layers are implemented as described in Kritchen (2023) and consist of two convolution layers and an 'add-in' layer of the original input. Generally, these maintain the input layer size on output. However an additional convolutional layer is applied to the 'Add-in' layer in the frog model at Identity Resnet layers marked with * in Table 3 to transform layer sizes.

Table 3. CNN model designs used for bat and frog models. Layers marked with * contain an extra convolutional layer for layer size reduction. CNN terms are from Kritchen (2023).

Bat model layers (neurons, kernel, stride)	Layer size	Frog model layers (neurons, kernel, stride)	Layer size
Input Layer	1124, 163	Input layer	70, 163
Conv1D (128, 3, 2)	560, 128	Conv1D (128, 3, 1)	68, 128
Conv1D (128,3,1)	558, 128	Identity Resnet Conv1D (128, 3, 1)	68, 128
Dropout 0.2		Identity Resnet Conv1D (128, 3, 1)	68, 128
Conv1D (129, 9,4)	138, 128	Identity Resnet Conv1D (128, 3, 2) *	34, 128
Conv1D (128, 5,2)	67, 128	Dropout 0.2	
MaxPooling 2x	33, 128	Conv1D (128, 3, 1)	32, 128
Conv1D (128, 3, 1)	31, 128	Identity Resnet Conv1D (128, 3, 1)	32, 128
MaxPooling 2x	15, 128	Identity Resnet Conv1D (128, 3, 1)	32, 128
Conv1D (128, 3,1)	13, 128	Identity Resnet Conv1D (128, 3, 2) *	16, 128
MaxPooling 2x	6, 128	Dropout 0.2	
Dropout 0.2		Conv1D (128, 3, 1)	14, 128
Conv1D (128, 3, 1)	4, 128	Identity Resnet Conv1D (128, 3, 1)	14, 128
Flatten	512	Identity Resnet Conv1D (128, 3, 2) *	7, 128
Dense (17) Softmax	17	Dropout 0.2	
		Conv1D (256, 3, 1)	5, 256
		Identity Resnet Conv1D (256, 3, 1)	5, 256
		Identity Resnet Conv1D (256, 3, 1)	5, 256
		Identity Resnet Conv1D (256, 3, 2) *	3, 256
		Conv1D (512, 3, 1)	1, 512
		Flatten	512
		Dense (17) Softmax	17

2.5 Model training data

The bat model training data was extracted from free-flight recordings of identified individuals from 16 species of bat that occur in southwest Victoria, excluding the atypical pulses recorded immediately after release. The 59,229 0.75-second exemplars used for the model were assembled from three sources with ARI contributing approximately 51% of the calls, NSW Department of Primary Industries and Regional Development contributing 28% and the University of Melbourne 21%. For species that display geographic variation in their calls (e.g. *Vespadelus* spp., Law et al., 2002) only calls from southwest Victoria were used, while calls from outside the region were included for some species without known geographic variation. The noise data was extracted from sections of the recordings that did not containing bat calls (Table 4). Noise data includes environmental sounds such as insects and sections of relative silence. The PreciseBatFiltered sampling strategy was selected (Table 2) for the bat model as the higher frame rate may provide precision required to differentiate species with similar calls.

The frog model training data consisted of expert-identified calls assembled from many years of recordings curated by ARI (Table 4). The model targets 15 frog species and a 'catch-all' class named 'Frog chorus'. This class covers the common instance of two or more frog species calling simultaneously and where there is a cacophony of calls making species identification difficult. This class provides a fall-back level of frog activity as simultaneous multi-species identification is not the purpose of a model trained with 'one-hot-vector' labels. Note that the frog model has an order of magnitude more noise exemplars than that of the bat model. This is due to the vast variety of lower frequency noise sounds that may be encountered. Some are extracted from FSD50K sound dataset (Fonseca et al., 2022) which contains a huge variety of anthropogenic sounds such as voices, music, vehicles and common household and farm sounds. Many exemplars are noises identified within ARI audio files such as wind, rain, sticks rubbing, and anthropogenic sounds such as planes and traffic. Exemplars of other native bird and mammal species contained within the ARI dataset are also included but reclassed as noise. The BirdFrogSlowAudio sampling strategy was selected (Table 2) for the frog model.

Table 4. The number of training and test exemplars used in the models for bats and frogs.

Species	Scientific name	Abbrev.	Training / Test
Bats			
White-striped Freetail Bat	*Austronomus australis*	Aa	2229 / 922
Gould's Wattled Bat	*Chalinolobus gouldii*	Cg	4185 / 1599
Chocolate Wattled Bat	*Chalinolobus morio*	Cm	738 / 372
Eastern False Pipistrelle	*Falsistrellus tasmaniensis*	Ft	1398 / 678
Southern Bent-wing Bat	*Miniopterus orianae bassanii*	Mob	1507 / 673
Eastern Bent-wing Bat	*Miniopterus orianae oceanensis*	Moo	1119 / 341
Large-footed Myotis	*Myotis macropus*	Mm	2056 / 758
Lesser Long-eared Bat	*Nyctophilus geoffroyi*	Nge	1686 / 731
Gould's Long-eared Bat	*Nyctophilus gouldi*	Ngo	334 / 146
Southern Freetail Bat	*Ozimops planiceps*	Op	2114 / 805
Eastern Freetail Bat	*Ozimops ridei*	Or	717 / 269
Yellow-bellied Sheathtail Bat	*Saccolaimus flaviventris*	Sf	773 / 396
Inland Broad-nosed Bat	*Scotorepens balstoni*	Sb	932 / 399
Large Forest Bat	*Vespadelus darlingtoni*	Vd	636 / 386
Southern Forest Bat	*Vespadelus regulus*	Vr	182 / 69
Little Forest Bat	*Vespadelus vulturnus*	Vv	529 / 202
Noise for bat models			20,592 / 8,752
Frogs			
Plains Froglet	*Crinia parinsignifera*	PF	1707 / 783
Common Froglet	*Crinia signifera*	CF	6036 / 2710
Sloane's Froglet	*Crinia sloanei*	SF	119 / 50
Victorian Smooth Froglet	*Geocrinia victoriana*	VSF	640 / 272
Giant Burrowing Frog	*Heleioporus australiacus*	GBF	333 / 145
Pobblebonk	*Limnodynastes dumerilii*	PF	381 / 157
Barking Marsh Frog	*Limnodynastes fletcheri*	BMF	894 / 385
Spotted Marsh Frog	*Limnodynastes tasmaniensis*	SMF	674 / 277
Common Spadefoot Toad	*Neobatrachus sudellae*	CST	364 / 147
Peron's Tree Frog	*Pengilleyia peronii*	PTF	2105 / 953
Dendy's Toadlet	*Pseudophryne dendyi*	DT	645 / 286
Southern Toadlet	*Pseudophryne semimarmorata*	ST	446 / 195
Growling Grass Frog	*Ranoidea raniformis*	GGF	696 / 292
Watson's Tree Frog	*Rawlinsonia watsoni*	WTF	1228 / 517
Southern Brown Tree Frog	*Rawlinsonia ewingii*	SBTF	514 / 201
Frog chorus		FC	2773 / 1017
Noise for frog models			247,988 / 105,468

2.6 Model training

Models are developed in Python 3.10 utilising TensorFlow 2.10 and Keras 2.10 (Chollet 2015). This runs on a Windows 11 computer with 256 Gb RAM, a 64-core Threadripper CPU and a Nvidia RTX3090 GPU. The training exemplars, in the form of their precomputed sample strategy matrices, and their labels, are queried from a MySQL 8.0 database. The models were trained using approximately 70% of the available exemplars and are assessed below against the remaining 30% holdout exemplars (Table 4). Audio samples that contributed more than one exemplar were not split between train and test datasets resulting in the slight variations of the 70-30 split. Both models were trained for 20 epochs using the categorial crossentropy loss function and Adam optimiser within Keras. The selection of the exemplars from the database and training of each model took approximately 30 minutes.

3 Results

For the bat model, an average accuracy of 90.5% of bat species in the 30% test holdout dataset were correctly identified. For the noise class, the accuracy is 99.7%. For the frog model, an average accuracy of 96.8% of frog species was achieved for the test dataset, excluding the noise class. The noise class accuracy of the frog model is 99.8%.

Table 5. The 70% training and 30% test holdout species accuracies for the bat and frog models.

Bat species	Training accuracy	Test accuracy	Frog species	Training accuracy	Test accuracy
White-striped Freetail Bat	99.0%	96.6%	Barking Marsh Frog	100.0%	95.3%
Gould's Wattled Bat	98.2%	93.9%	Common Froglet	99.6%	98.0%
Chocolate Wattled Bat	99.7%	88.4%	Common Spadefoot Toad	99.2%	95.2%
Eastern False Pipistrelle	99.7%	98.1%	Dendy's Toadlet	100.0%	98.2%
Southern Bent-wing Bat	94.4%	86.0%	Plains Froglet	99.6%	94.1%
Eastern Bent-wing Bat	92.1%	81.6%	Giant Burrowing Frog	97.0%	97.3%
Large-footed Myotis	99.4%	97.2%	Growling Grass Frog	100.0%	100.0%
Lesser Long-eared Bat	99.3%	92.7%	Watson's Tree Frog	100.0%	98.9%
Gould's Long-eared Bat	75.4%	40.4%	Peron's Tree Frog	98.6%	95.5%
Southern Freetail Bat	98.9%	91.5%	Pobblebonk Frog	99.7%	94.3%
Eastern Freetail Bat	97.8%	79.2%	Sloane's Froglet	96.6%	94.0%
Yellow-bellied Sheathtail Bat	100.0%	99.7%	Southern Brown Tree Frog	99.6%	99.0%
Inland Broad-nosed Bat	99.2%	82.2%	Southern Toadlet	99.6%	99.0%
Large Forest Bat	93.6%	77.2%	Spotted Marsh Frog	99.0%	89.2%
Southern Forest Bat	83.0%	62.3%	Victorian Smooth Froglet	99.8%	97.4%
Little Forest Bat	98.7%	93.0%	Frog chorus	99.6%	96.5%
Noise	99.9%	99.7%	Noise	99.9%	99.8%

Due to the large and arbitrary number of noise exemplars used in each model, traditional precision, recall and accuracy statistics are skewed by the zero-class exemplar counts. For the 30% test holdout data, it is far more informative to examine the confusion matrices. The confusion matrix for the bat model (Table 6) indicates that calls of Gould's Wattled Bat (*Chalinolobus morio*) are the most likely to be confused with a variety of other bat species. Other confused species are discussed below.

For the frog model, the confusion matrix (Table 7) indicates that the catch-all class 'Frog chorus' is most likely to be confused across a variety of species. However, this is to be expected to some degree, as this class contains the calls of other species and reflects the short-comings of the '1-hot-vector' approach for frog choruses.

Table 6. The confusion matrix of the 30% test holdout exemplars for the bat model. Observed species (row) versus predicted (column). See Table 4 for species name abbreviations.

	Aa	Cg	Cm	Ft	Mob	Moo	Mm	Nge	Ngo	Op	Or	Sf	Sb	Vd	Vr	Vv	Noise
Aa	890	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32
Cg	0	1502	0	0	0	0	3	0	2	23	8	0	61	0	0	0	0
Cm	0	0	329	0	9	2	0	3	1	2	0	0	2	0	1	13	10
Ft	0	0	0	665	0	0	0	5	0	0	0	0	3	2	0	0	3
Mob	0	0	9	0	579	31	0	2	0	0	0	0	0	0	0	47	5
Moo	0	0	2	0	12	278	0	4	10	0	0	0	0	0	4	28	3
Mm	0	1	0	3	0	0	737	13	1	1	0	0	2	0	0	0	0
Nge	0	4	0	15	0	0	4	678	18	2	0	0	2	1	0	1	6
Ngo	0	1	0	0	0	0	1	80	59	0	0	0	0	0	0	0	5
Op	0	43	0	0	0	0	8	0	0	736	7	0	9	0	0	0	2
Or	0	14	0	1	0	0	0	0	0	4	213	0	37	0	0	0	0
Sf	0	0	0	0	0	0	0	0	0	0	0	395	0	0	0	0	1
Sb	0	34	0	0	0	0	2	0	0	35	0	0	328	0	0	0	0
Vd	0	0	0	34	0	9	0	19	3	0	0	0	0	298	18	3	2
Vr	0	0	0	0	0	1	0	14	3	0	0	0	0	6	43	1	1
Vv	0	0	3	0	4	2	0	1	0	0	0	0	0	0	0	188	4
Noise	9	1	0	3	0	0	3	3	0	3	2	2	2	0	0	1	8723

Table 7. The confusion matrix of the 30% test holdout exemplars for the frog model. Observed species (row) versus predicted (column). See Table 4 for species name abbreviations.

	BMF	CF	CST	DT	ESB	FC	GBF	GGF	WTF	PTF	PF	SF	SBTF	ST	SMF	VSF	Noise
BMF	367	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	15
CF	0	2657	0	1	11	19	0	0	0	0	0	3	1	0	0	0	18
CST	1	0	140	0	0	3	0	0	0	0	0	0	0	0	0	0	3
DT	0	2	0	281	0	0	0	0	0	0	0	0	3	0	0	0	0
ESB	0	5	0	0	737	13	0	0	0	1	0	0	0	0	0	0	27
FC	0	1	9	0	19	981	0	0	0	2	0	0	0	0	1	0	4
GBF	0	0	0	0	0	0	141	0	0	0	0	0	0	0	0	0	4
GGF	0	0	0	0	0	0	0	292	0	0	0	0	0	0	0	0	0
WTF	0	0	0	0	0	0	0	0	511	0	0	0	0	0	0	0	6
PTF	1	5	3	0	0	14	0	0	0	910	0	0	0	0	2	0	18
PF	0	0	0	0	0	0	0	0	0	0	148	0	0	0	0	0	9
SF	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	3
SBTF	0	1	0	0	0	0	0	0	0	0	0	0	199	0	0	1	0
ST	0	0	0	1	0	0	0	0	0	0	0	0	0	193	0	1	0
SMF	2	0	0	1	0	10	0	1	0	2	0	0	0	0	247	0	14
VSF	0	0	0	0	1	0	0	0	0	0	0	0	3	1	0	265	2
Noise	34	6	1	0	52	5	0	23	5	12	34	0	2	1	25	1	105267

3.1 Production software

Our custom model application software, ARI Species Acoustics (ARISA), allows the user to select the model and audio files to be batch processed by directory, where each directory may contain many files. Each file may be seconds to hours long and may be recorded at differing sampling rates. ARISA loads not only the model, but

also the instructions gleaned from the sampling strategy JSON file, on how to correctly supply the data to the model. ARISA produces a comma separated text file (CSV file) that reports on which files contain target species calls and the time(s) within each recording at which they were detected. A vector of probabilities across all the model classes is also given. The software processes many files simultaneously, up to hundreds if they are brief, provided the computer has sufficient processing power. This software can process audio files at a rate of 300 seconds/second on a powerful workstation. Thus, one year's worth of continuous recordings may be processed in little more than one day.

The resultant CSV file, which may be subsetted by the user to sites, species or times of interest, can then be examined using the ARI Ecological Listener (ARIEL) validation software (Francis and Griffioen, 2024). This software steps through each identification, displaying the corresponding spectrogram, the model probability estimates, allows the user to listen to the call identified within the original sound file, and most crucially, provide the capacity for the user to annotate the call identified. The user may accept or reject the identification and in doing so, provide a new label for the sound encountered. This feedback is stored in a validation CSV file which may be used for post analysis, reporting, or to improve the model with new training data for the next generation.

4 Discussion

The results indicate that given high-quality training data, the 1-D CNN produced with these designs and sampling strategies produce excellent results. Of note is the very low false positive error rate for the noise category. This has significant implications in the utility of the models for identifying species within field data. Models should include a 'noise' or 'non-target' class, for without such a class and significant preprocessing, there would likely be an abundance of false positives that would undermine the model's application for field data. In practice, most of the post-model iterative tuning is done by the identification of noise sources that generate false positives and resupplying these as 'non-target' exemplars.

The models performed well on both the training and independent test dataset. As to be expected, the models overfitted the training data compared to the test holdout data (see Table 5). However, the bat model performed exceptionally well on the test dataset. The worst performing bat species, Gould's Long-eared Bat, cannot be reliably distinguished from Lesser Long-eared Bats by experts from spectrograms, so similar are their calls (Lindy Lumsden pers. obs.). Interestingly, another species with similar calls, the Large-footed Myotis, which is often combined with the long-eared bats into a species complex, had a high positive identification rate (97%). The frog model performed well across all species. For both models, classification accuracies of species with fewer exemplars were generally lower, highlighting the need for a large number of calls to be included to train these models.

The confusion matrices of both models (Tables 6 and 7) point to which training data deserves further examination. Some false positives are expected, such as the Gould's and Lesser Long-eared Bats, and Frog choruses with individual species, but others may indicate a problem with mis-labelling or poor exemplars. For example, the White-striped Freetail Bat typically has a longer time between calls than other bats, often spaced at one second between calls. The high noise misclassification for this species may indicate that some of the 0.75-second labelled exemplars fell between actual calls, and hence only contained noise. In addition, further testing needs to be undertaken on field-collected bat calls, which are likely to be of lower quality than the exemplars used to train and test the model, with the learnings fed back into the model to refine it further.

In most cases it is preferrable that common species are incorrectly classified as noise (false negatives) rather than being mis-identified as other species (false positives). Large numbers of false positives make the validation process difficult even with the efficient ARIEL software. If common misidentifications are readily identified, they usually can be corrected through retraining of the model. Conversely, if rare species are sought, false positives may be tolerated at a reasonable level given the ARIEL software's ability to rapidly validate hundreds of calls per hour.

The fusion of spectrographic and other audio indices within the one model is a major advantage of our 1-D design. Similar to side and front elevations of house designs, this approach offers different views of the same data. Adding other indices, be they derived from musical characterisation, speech analysis, or signal processing, can easily be accommodated provided the indices are relevant at the frame size.

The merits of layer designs of CNN models are often discussed (Kritchen 2023, Carvalho et al., 2021). While this is essential to progress the technology, experience has shown that provided the designs are well configured, the performance differences may not be large between designs. Many CCN designs are available for

selection within our system, varying from 7 to 30 convolutional layers and with 500,000 to 4,000,000 parameters. Different designs seem to be sensitive to different sound characteristics for some species and noise within the training data. For example, another bat model with 1,150,000 parameters trained with the exact same 70% training data produced the slightly lesser species accuracy of 89.9% compared to the model presented (90.5%). However, if this model is average-ensembled with the model presented, the ensemble produces an average species accuracy of 92.0% for the test data. As a result, models destined for processing field data produced by our system are always ensembles of 3 to 5 models, of varying designs and each trained with folds of 70% of the total data available. These ensembles should perform at least as well as, and most likely better, than the component models. However, as these mixed-fold ensembles have 'seen' all the training data, their holdout statistics are compromised and only the component models statistics can be cited. Ensembles incur only a modest computational cost as TensorFlow efficiently ensembles the models and applies them on the computer's GPU.

5 Conclusion

Our 1-D CNN approach combined with tailored sampling strategies has streamlined our production of acoustic models and their application. The production environment is managed by a database, which collates field audio files, species observations, sampling strategies, CNN designs, model class lists, model configuration and model fit statistics. Custom models may be configured and batched in minutes and produced without intervention after that. This has greatly sped up the investigation into data sampling strategies and model designs and the creation of custom models, such as is required for a 22.05 kHz Koala dataset recorded in NSW. The combination of spectrographic and other sound indices within the one CNN design is easily implemented. Furthermore, the accuracy of this classification system may be improved through the augmentation of more descriptive audio indices in the future.

Sampling strategies support the customisation of the models to reflect the target species, the training data available and the target dataset to be analysed. The chosen strategy links the model design to its application in ARISA. ARIEL efficiently facilitates validation of detections made by ARISA to meet the researcher's needs. It also provides feedback to improve training datasets. The integration of this processing system makes iterative model improvement a defining feature. As any audio classification model is only as good as the data used to build it, improving the data quality can provide significant accuracy gains, complementing the novel designs presented here. This system embodies the principle that the best model is often your last model.

ACKNOWLEDGEMENTS

We thank Amanda Bush and Pia Lentini (ARI), Brad Law and Leroy Gonsalves (NSW DPI) and Ophelie Planckaert (University of Melbourne) and for collecting and preparing bat calls for inclusion in the bat model, and Katie Howard (ARI), Matt West (Wild Research) and Greg Clarke (Wildlife Unlimited) for assisting with frog call annotation. We also thank Nevil Amos and Frank Amtstaetter (ARI) for comments on an earlier draft of this manuscript.

REFERENCES

Abdoli, S., Cardinal, P., Koerich, A. (2019). 'End-to-end environmental sound classification using a 1D Convolutional Neural Network'. *arXiv: 1904.08990*. https://arxiv.org/abs/1904.08990

Allamy, S., Koerich, A. (2021). '1D CNN architectures for music genre classification'. *arXiv: 2105.07302*. https://arxiv.org/abs/2105.07302

Carvalho, S., Gomes, E.F. (2021). 'Automatic identification of bird species from audio'. *Intelligent Information and Database Systems*. ACIIDS 2021. Lecture Notes in Computer Science, vol 12672. https://doi.org/10.1007/978-3-030-73280-6_4

Chollet, F. (2015). 'Keras' [Online]. https://github.com/fchollet/keras

Eichinski, P., Alexander, C., Roe, P., Parsons, S., Fuller, S. (2022). 'A Convolutional Neural Network Bird Species Recognizer built from little data by iteratively training, detecting, and labeling'. *Front. Ecol. Evol.,* l. 10:810330. doi: 10.3389/fevo.2022.810330 https://doi.org/10.1002/ece3.8873

Fonseca, E., Favory, X., Pons, J., Font, F., Serra, X. (2022). 'FSD50K: an open dataset of human-labeled sound events'. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. Vol 30, pp 829-852. IEEE

Francis, L. and Griffioen, P. (2024). ARIEL: Arthur Rylah Institute Ecoacoustic Listener. Flexible, open-sourced software for the rapid validation of acoustic data in your workflow, Melbourne. Zenodo. https://doi.org/10.5281/zenodo.10681701

Giannakopoulos, T. (2015). 'pyAudioAnalysis: An open-source Python library for audio signal analysis'. *PLoS ONE* 10(12): e0144610. https://doi.org/10.1371/journal.pone.0144610

He, K., Zhang, X., Ren, S. and Sun, J. (2015). 'Deep Residual Learning for Image Recognition'. *arXiv:1512.03385 [cs.CV]*, https://doi.org/10.48550/arXiv.1512.03385

Himawan, I., Towsey, M., Law, B., & Roe, P. 2018. 'Deep Learning Techniques for Koala Activity Detection'. *Interspeech 2018*, 2107–2111. https://doi.org/10.21437/interspeech.2018-1143

Kahl, S,,Wood, C., Eibl, M., Klink, H. (2021). 'BirdNET: A deep learning solution for avian diversity monitoring'. *Ecological Informatics , 61, 101236.* https://www.sciencedirect.com/science/article/pii/S1574954121000273

Krichen, M. (2023). 'Convolutional Neural Networks: A Survey'. *Computers,* 12(8), 151. https://doi.org/10.3390/computers12080151

Law, B. S., Reinhold, L., Pennay, M. (2002). 'Geographic variation in the echolocation calls of *Vespadelus* spp. (Vespertilionidae) from New South Wales and Queensland, Australia.' *Acta Chiropterologica* **4**(2): 201-215.

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'. *Ecological Informatics*, 59, 101113. https://doi.org/10.1016/j.ecoinf.2020.101113

McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E., Nieto, O. (2015). "librosa: Audio and music signal analysis in python." In Proceedings of the 14th Python in Science Conference, pp. 18-25. 2015.

Nanni, L., Costa, Y. M. G., Aguiar, R. L., Mangolin, R. B., Brahnam, S., Silla Jr., C. N., (2020). 'Ensemble of convolutional neural networks to improve animal audio classification'. *Journal on Audio, Speech, and Music Processing,* 8 (2020). https://doi.org/10.1186/s13636-020-00175-3

Piczak, K. J. (2016). 'Recognizing bird species in audio recordings using deep convolutional neural networks'. *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016. Volume 1609 of CEUR Workshop Proceedings,* pages 534-543, CEUR-WS.org, 2016.

Ruff, Z. J., Lesmeister, D. B., Duchac, L. S., Padmaraju, B. K., Sullivan, C. M. (2019). 'Automated identification of avian vocalizations with deep convolutional neural networks'. *Remote Sensing in Ecology and Conservation*, 6(1), 79–92. https://doi.org/10.1002/rse2.125

Sharan, R. V., Xiong, H., Berkovsky, S. (2021). 'Benchmarking audio signal representation techniques for classification with convolutional neural networks'. *Sensors*, *21*(10), Article 3434. https://doi.org/10.3390/s21103434

Xie, J., Zhu, M., Hu, K., Zhang, J., Hines, H., Guo, Y. (2022). 'Frog calling activity detection using lightweight CNN with multi-view spectrogram: A case study on Kroombit tinker frog'. *Machine Learning with Applications*, Vol 7, 2022,100202, https://doi.org/10.1016/j.mlwa.2021.100202.

Zhong, M., LeBien, J., Campos-Cerqueira, M., Dodhia, R., Lavista Ferres, J., Velev, J. P., Aide, T. M. (2020). 'Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling'. *Applied Acoustics*, 166, 107375. https://doi.org/10.1016/j.apacoust.2020.107375