gibbonNetR: an R Package for the Use of Convolutional Neural Networks for Automated Detection of Acoustic Data

A Preprint

Dena Jane Clink
K. Lisa Yang Center for Conservation Bioacoustics,Cornell Lab of Ornithology
Cornell University
Ithaca, New York, United States
dena.clink@cornell.edu

Abdul Hamid Ahmad
Institute for Tropical Biology and Conservation
Universiti Malaysia Sabah (UMS)
Kota Kinabalu, Sabah, Malaysia
abdulhamidahmad.sabah@gmail.com

February 15, 2025

Abstract

Automated detection of acoustic signals is crucial for effective monitoring of vocal animals and their habitats across large spatial and temporal scales. Recent advances in deep learning have made high performance automated detection approaches accessible to more practitioners. However, there are few deep learning approaches that can be implemented natively in R. The 'torch for R' ecosystem has made the use of convolutional neural networks (CNNs) accessible for R users. Here, we provide an R package and workflow to use CNNs for automated detection and classification of acoustics signals from passive acoustic monitoring data. We provide examples using data collected in Sabah, Malaysia. The package provides functions to create spectrogram images from labeled data, compare the performance of different CNN architectures, deploy trained models over directories of sound files, and extract embeddings from trained models. The R programming language remains one of the most commonly used languages among ecologists, and we hope that this package makes deep learning approaches more accessible to this audience. In addition, these models can serve as important benchmarks for future automated detection work.

***K*eywords** deep learning · passive acoustic monitoring · gibbon · automated detection

1 Statement of need

1.1 *Passive acoustic monitoring*

We are in a biodiversity crisis, and there is a great need for the ability to rapidly assess biodiversity in order to understand and mitigate anthropogenic impacts. One approach that can be especially effective for monitoring of vocal yet cryptic animals is the use of passive acoustic monitoring (Gibb et al. 2018), a technique that relies on autonomous acoustic recording units. PAM allows researchers to monitor vocal animals and their habitats at temporal and spatial scales that are impossible to achieve using only human observers. Interest in use of PAM in terrestrial environments has increased substantially in recent years (Sugai et al. 2019), due to reduced price of the recording units and improved battery life and data storage capabilities. However, the use of PAM often leads to the collection of terabytes of data that is time- and cost-prohibitive to analyze manually.

1.2 *Automated detection*

Automated detection for PAM data refers to identifying the start and stop time of signals of interest within a longer sound recording (Stowell 2022). Some of the early non-deep learning approaches for the automated detection of acoustic signals in terrestrial PAM data include binary point matching (Katz, Hafner, and Donovan 2016), spectrogram cross-correlation (Balantic and Donovan 2020), or the use of a band- limited energy detector and subsequent classifier, such as support vector machine (Clink et al. 2023; Kalan et al. 2015). Recent advances in deep learning have revolutionized image and speech recognition (LeCun, Bengio, and Hinton 2015), with important cross-over for the analysis of PAM data. Traditional approaches to machine learning relied heavily on feature engineering, since early machine learning algorithms required a reduced set of representative features that were chosen by researchers, such as features estimated from the spectrogram.

Deep learning does not require feature engineering (Stevens, Antiga, and Viehmann 2020), as the algorithms include a step that identifies relevant features from the input. This can lead to faster development time and increased ability to represent complex patterns typically seen in image and acoustic data. Convolutional neural networks (CNNs) — one of the most widely used deep learning algorithms—are useful for processing data that have a 'grid-like topology', such as image data that can be considered a 2-dimensional grid of pixels (Goodfellow, Bengio, and Courville 2016). The 'convolutional' layer learns the feature representations of the inputs; these convolutional layers consist of a set of filters which are basically two-dimensional matrices of numbers and the primary parameter is the number of filters (Gu et al. 2018). If training data are scarce, overfitting may occur as representations of images tend to be large with many variables (LeCun, Bengio, and others 1995).

1.3 *Transfer learning*

Training deep learning models generally requires a large amount of training data and substantial computing resources, which. Transfer learning is an approach wherein the architecture of a pretrained CNN (which is generally trained on a very large dataset) is applied to a new classification problem. For example, CNNs trained on the ImageNet dataset of > 1 million images (Deng et al. 2009) such as ResNet have been applied to automated detection/classification of primate and bird species from PAM data (Dufourq et al. 2022; Ruan et al. 2022). Generally, very few practitioners train a CNN from scratch, and there are two common approaches for transfer learning. The first option is to use the CNN as a feature extractor, and train only the last classification layer. The second option is known as 'fine-tuning', where instead of initializing a neural network with random weights, initialization is done using the pre-trained network. Using these pre-trained weights are valuable because the model has already learned useful feature representations (Takhirov 2021). Both approaches require substantially less computing power than training from scratch. The functions in the 'gibbonNetR' package allow users to train models with both types of transfer learning.

1.4 State of the field

The two most popular open-source programming languages are R and Python (Scavetta and Angelov 2021). Python has surpassed R in terms of overall popularity, but R remains an important language for the life sciences (Lawlor et al. 2022). 'Keras' (Chollet and others 2015), 'PyTorch' (Paszke et al. 2019) and 'Tensorflow' (Martín Abadi et al. 2015) are some of the more popular neural network libraries; these libraries were all initially developed for the Python programming language. One of the earliest implementations of automated detection using R was the 'monitoR' package, that included functions for template detection (Katz, Hafner, and Donovan 2016). The 'warbleR' package included functions for energy-based detection, which identifies signals of interest in a certain frequency range above specified energy thresholds (Araya-Salas and Smith-Vidaurre 2017). The 'gibbonR' package combined energy-based detection with tranditional machine learning classification (Clink and Klinck 2019).

Until recently, deep learning implementations in R relied on the 'reticulate' package which served as an interface to Python (Ushey, Allaire, and Tang 2022). Previous implementations of automated detection using deep learning in R relied on the 'reticulate' package Silva et al. (2022). However, the recent release of the 'torch for R' ecosystem provides a framework based on 'PyTorch' that runs natively in R and has no dependency on Python (Falbel 2023). Running natively in R means more straightforward installation, and higher accessibility for users of the R programming environment. Keydana (2023) provides tutorials for transfer learning in the 'torch for R' ecosystem, and the functions in 'gibbonNetR' rely heavily on these tutorials. Variations of the transfer learning approaches included in this package have already been implemented in Python (Dufourq et al. 2022). Recent advances have used embeddings from audio classification models trained on bird songs for new classification problems, and in most cases these embeddings led to better performance than general audio or image datasets (Ghani et al. 2023).

2 Overview

This package provides functions to create spectrogram images using the 'seewave' package (J. Sueur, T. Aubin, and C. Simonis 2008), use transfer learning for six CNN architectures: AlexNet (Krizhevsky, Sutskever, and Hinton 2017) , VGG16, VGG19 (Simonyan and Zisserman 2014), ResNet18, ResNet50, and ResNet152 (He et al. 2016)) trained on the ImageNet dataset (Deng et al. 2009). This package has been used for automated detection of gunshots (Vu et al. 2024) and gibbon calls (Clink, Kim, et al. 2024; Clink, Cross-Jaya, et al. 2024). The package also has functions to evaluate model performance, deploy the highest performing model over a directory of sound files, and extract embeddings from trained models to visualize acoustic data. We provide an example dataset that consists of labelled vocalizations of the loud calls of four vertebrates (see detailed description below) from Danum Valley Conservation Area, Sabah, Malaysia (Clink and Hamid Ahmad 2024). Detailed usage instructions for 'gibbonNetR' can be found Here

2.1 Data summary

We include sound files and spectrogram images of five sound classes: great argus pheasant (*Argusianus argus*) long calls (Clink et al. 2021), helmeted hornbills (*Rhinoplax vigil*), and rhinoceros hornbills (*Buceros rhinoceros*) (Kennedy et al. 2023), female gibbons (*Hylobates funereus*) and a catch-all "noise" category. The data come from two separate PAM arrays in Danum Valley Conservation Area, Sabah, Malaysia. The training and validation data come from a wide array of Swift autonomous recording units placed on ~750 m spacing (Clink et al. 2023), and the test data come from a different, smaller array (~250 m spacing) within the same area. We used a band-limited energy detector to identify signals that were 3-sec or longer duration within the 400-1600 Hz range, and then a single observer (DJC) manually sorted the detections into their respective categories (Clink et al. 2023).

2.2 Preparing training, validation, and test data

The package currently uses spectrogram images (Figure 1) to train and evaluate CNN model performance, and we includes a function that can be used to create spectrogram images from Waveform Audio File Formant (.wav) files. The .wav files should be organized into separate folders, with each folder named according to the class label of the files it contains. We highly recommend that your test data come from a different recording time and/or location to better understand the generalizability of the models (Stowell 2022).

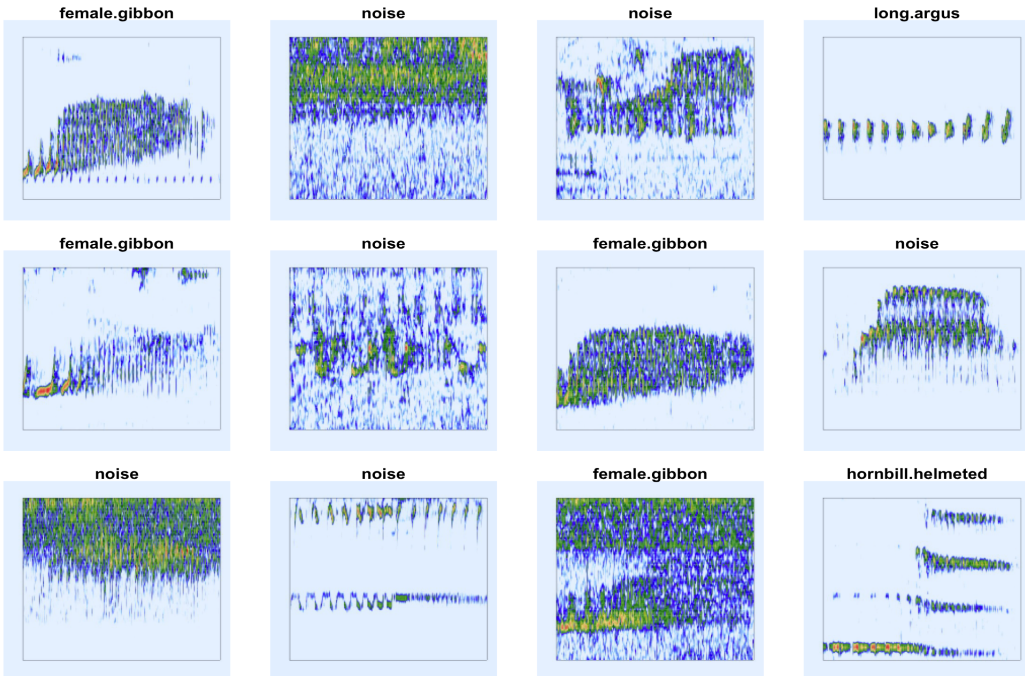

Figure 1: Spectrograms of training clips for CNNs

2.3 Model training

The package currently allows for the training of six different CNN architectures ('alexnet', 'vgg16', 'vgg19', 'resnet18', 'resnet50', or 'resnet152'), and the user can specify if they want to freeze the feature extraction layers or not. There is the option to train a binary or multi-class classifer.

2.4 Evaluate model performance

We can compare the performance of different CNN architectures (Figure 2). Using the 'get_best_performance' function we can evaluate the performance of different model architectures on the test dataset for the specified class. We can calculate the best F1, precision, recall using the 'caret' package (Kuhn 2008), and the area under the ROC (Receiver Operating Characteristic) curve using the 'ROCR' package (Sing et al. 2005), which evaluates the classifier's ability to discriminate between positive and negative classes.

```
PerformanceOutput <- gibbonNetR::get_best_performance(performancetables.dir=performancetables.dir,
                                                      class='female.gibbon',
                                                      model.type = "multi",Thresh.val=0)
PerformanceOutput$f1_plot
```

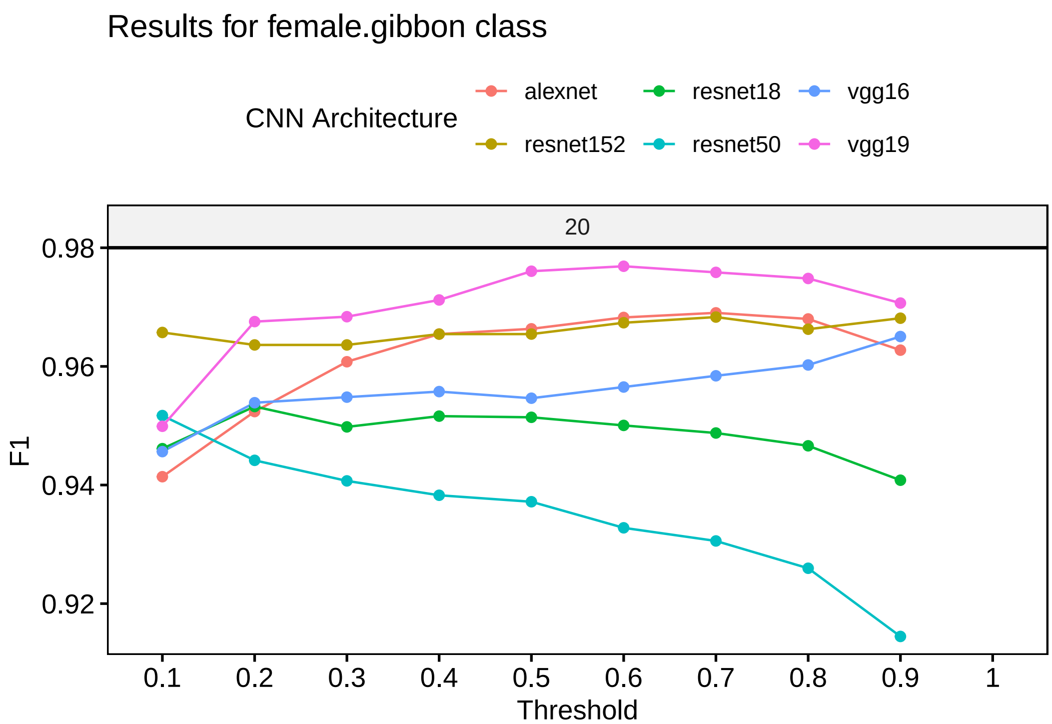

Figure 2: Evaluating performance of pretrained CNNs

2.5 Extract embeddings

Embeddings from deep learning models can be used as features in unsupervised approaches, with promising results for call repertoires (Best et al. 2023) and individual identity (Lakdari et al. 2024). This package contains a function to use pretrained CNNs to extract embeddings, where the trained model path, along with test data location and target class are specified. Depending on the research question, this output could be used to sort between true and false positives for automated detection, or to explore differences in call types or potential number of individuals in the dataset.

2.6 We can plot the unsupervised clustering results

In Figure 3 the top plot is a Uniform Manifold Approximation and Projection (UMAP) where each point represents one call, and the colors indicate the original class label. The bottom plot is the same UMAP plot, but with points colored based on cluster assignment by the 'hdbscan' algorithm (Hahsler, Piekenbrock, and Doran 2019).

```
#> processing embeddings
#> Unupervised clustering for female.gibbon
```

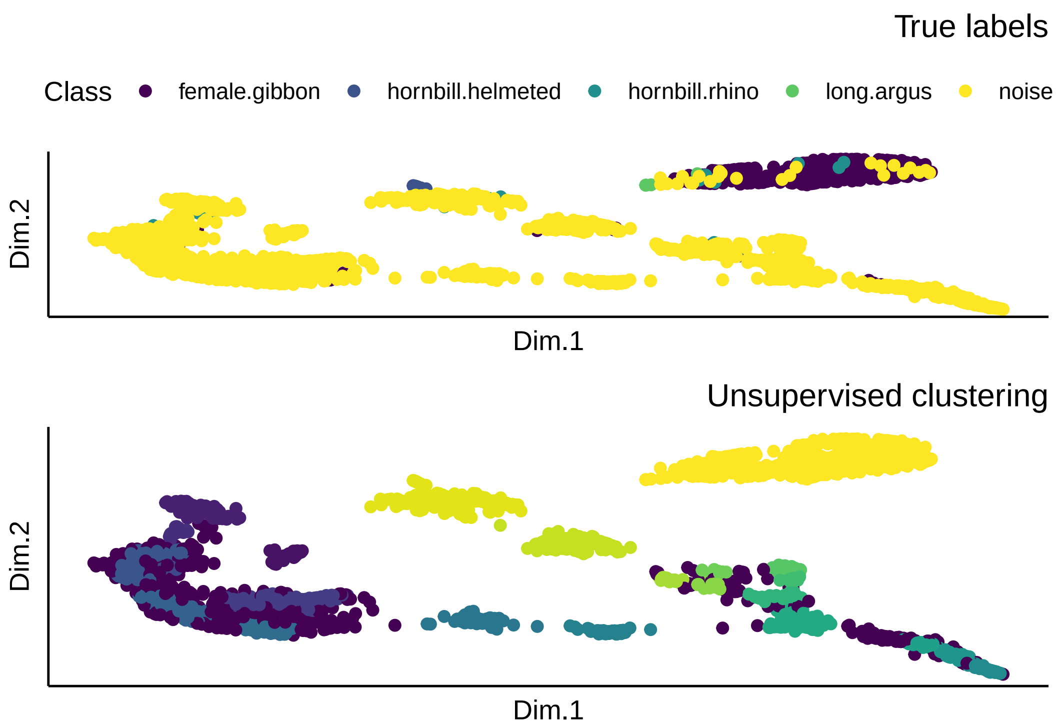

Figure 3: UMAP plot of embeddings from test data set colored by actual label (top) and unsupervised cluster assignment (bottom)

2.6.1 We can explore the unsupervised clustering results

We can calculate the Normalize Mutual Information score, which provides a value between 0 and 1, indicating the match between cluster labels and actual labels. We also create a confusion matrix using the 'caret' package (Kuhn 2008) which returns the results when we use the unsupervised clustering algorithm function 'hdbscan' (Hahsler, Piekenbrock, and Doran 2019) to match the target class to the cluster with the largest number of observations of that particular class.

3 Future directions

There have been huge advances in the fields of deep learning and automated detection for PAM data in recent years. The approach presented in this package is one of the first to use the 'torch for R' ecosystem and to employ automated detection using deep learning natively in R. More recent approaches use transfer learning from models that are explicitly trained on bioacoustics data, such as BirdNET (Ghani et al. 2023), have been introduced. There is a huge need in the field of bioacoustics to do benchmarking, wherein different model architectures and performance are compared across diverse datasets. The methods presented here can provide important benchmarks for future work, and for understanding how and if deep learning advances improve performance over more traditional methods. In addition, this package provides a comprehensive suite of tools for processing, analyzing, and visualizing acoustic data, providing robust support for tasks such as automated detection, feature extraction, classification, and data visualization, which are critical for conservation work using PAM. The R package is available on Github, where issues can be opened.

4 Acknowledgments

We would like to thank the Sabah Biodiversity Centre and Danum Valley Conservation Area for granting us permission to conduct research. We are incredibly grateful for the detailed comments provided by Steffi LaZerte, which substantially improved the package and documentation.

References

Araya-Salas, Marcelo, and Grace Smith-Vidaurre. 2017. "warbleR: An r Package to Streamline Analysis of Animal Acoustic Signals." *Methods in Ecology and Evolution* 8 (2): 184–91.

Balantic, Cathleen, and Therese Donovan. 2020. "AMMonitor: Remote Monitoring of Biodiversity in an Adaptive Framework with r." *Methods in Ecology and Evolution* 11 (7): 869877.

Best, Paul, Sébastien Paris, Hervé Glotin, and Ricard Marxer. 2023. "Deep Audio Embeddings for Vocalisation Clustering." *PLOS ONE* 18 (7): 1–18. `https://doi.org/10.1371/journal.pone.0283396`.

Chollet, François, and others. 2015. "Keras." `https://github.com/fchollet/keras`.

Clink, D. J., Hope Cross-Jaya, Jinsung Kim, Abdul Hamid Ahmad, Moeurk Hong, Roeun Sala, Hélène Birot, et al. 2024. "Benchmarking for the Automated Detection and Classification of Southern Yellow-Cheeked Crested Gibbon Calls from Passive Acoustic Monitoring Data." *bioRxiv*. `https://doi.org/10.1101/2024.08.17.608420`.

Clink, D. J., Tom Groves, Abdul Hamid Ahmad, and Holger Klinck. 2021. "Not by the Light of the Moon: Investigating Circadian Rhythms and Environmental Predictors of Calling in Bornean Great Argus." *Plos One* 16 (2): e0246564. 10.1371/journal.pone.0246564.

Clink, D. J., and Abdul Hamid Ahmad. 2024. "A Labelled Dataset of the Loud Calls of Four Vertebrates Collected Using Passive Acoustic Monitoring in Malaysian Borneo," November. `https://doi.org/10.5281/zenodo.14213067`.

Clink, D. J., Isabel Kier, Abdul Hamid Ahmad, and Holger Klinck. 2023. "A Workflow for the Automated Detection and Classification of Female Gibbon Calls from Long-Term Acoustic Recordings." *Frontiers in Ecology and Evolution* 11. `https://doi.org/10.3389/fevo.2023.1071640`.

Clink, D. J., Jinsung Kim, Hope Cross-Jaya, Abdul Hamid Ahmad, Moeurk Hong, Roeun Sala, Hélène Birot, et al. 2024. "Automated Detection of Gibbon Calls from Passive Acoustic Monitoring Data Using Convolutional Neural Networks in the" Torch for r" Ecosystem." *arXiv Preprint arXiv:2407.09976*.

Clink, D. J., and Holger Klinck. 2019. "gibbonR: An r Package for the Detection and Classification of Acoustic Signals." *arXiv Preprint arXiv:1906.02572*.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. "Imagenet: A Large-Scale Hierarchical Image Database." In, 248255. Ieee. 10.1109/cvpr.2009.5206848.

Dufourq, Emmanuel, Carly Batist, Ruben Foquet, and Ian Durbach. 2022. "Passive Acoustic Monitoring of Animal Populations with Transfer Learning." *Ecological Informatics* 70: 101688. `https://doi.org/10.1016/j.ecoinf.2022.101688`.

Falbel, Daniel. 2023. *Luz: Higher Level 'API' for 'Torch'*. `https://CRAN.R-project.org/package=luz`.

Ghani, Burooj, Tom Denton, Stefan Kahl, and Holger Klinck. 2023. "Global Birdsong Embeddings Enable Superior Transfer Learning for Bioacoustic Classification." *Scientific Reports* 13 (1): 22876. 10.1038/s41598-023-49989-z.

Gibb, Rory, Ella Browning, Paul Glover-Kapfer, and Kate E. Jones. 2018. "Emerging Opportunities and Challenges for Passive Acoustics in Ecological Assessment and Monitoring." *Methods in Ecology and Evolution*, October. `https://doi.org/10.1111/2041-210X.13101`.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, et al. 2018. "Recent Advances in Convolutional Neural Networks." *Pattern Recognition* 77: 354377.

Hahsler, Michael, Matthew Piekenbrock, and Derek Doran. 2019. "dbscan: Fast Density-Based Clustering with R." *Journal of Statistical Software* 91 (1): 1–30. `https://doi.org/10.18637/jss.v091.i01`.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." In, 770778. 10.1109/cvpr.2016.90.

J. Sueur, T. Aubin, and C. Simonis. 2008. "Seewave: A Free Modular Tool for Sound Analysis and Synthesis." *Bioacoustics* 18: 213–26.

Kalan, Ammie K., Roger Mundry, Oliver J J Wagner, Stefanie Heinicke, Christophe Boesch, and Hjalmar S. Kühl. 2015. "Towards the Automated Detection and Occupancy Estimation of Primates Using Passive Acoustic Monitoring." *Ecological Indicators* 54 (July 2015): 217226. `https://doi.org/10.1016/j.ecolind.2015.02.023`.

Katz, Jonathan, Sasha D Hafner, and Therese Donovan. 2016. "Assessment of Error Rates in Acoustic Monitoring with the r Package monitoR." *Bioacoustics* 25 (2): 177196. 10.1080/09524622.2015.1133320.

Kennedy, Amy G, Abdul Hamid Ahmad, Holger Klinck, Lynn M Johnson, and D. J. Clink. 2023. "Evidence for Acoustic Niche Partitioning Depends on the Temporal Scale in Two Sympatric Bornean Hornbill Species." *Biotropica* 55 (2): 517–28. 10.1111/btp.13205.

Keydana, Sigrid. 2023. *Deep Learning and Scientific Computing with r Torch*. CRC Press. 10.1201/9781003275923.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2017. "Imagenet Classification with Deep Convolutional Neural Networks." *Communications of the ACM* 60 (6): 8490. 10.1145/3065386.
Kuhn, Max. 2008. "Caret Package." *Journal of Statistical Software* 28 (5): 126.
Lakdari, Mohamed Walid, Abdul Hamid Ahmad, Sarab Sethi, Gabriel A Bohn, and D. J. Clink. 2024. "Mel-Frequency Cepstral Coefficients Outperform Embeddings from Pre-Trained Convolutional Neural Networks Under Noisy Conditions for Discrimination Tasks of Individual Gibbons." *Ecological Informatics* 80: 102457. 10.1016/j.ecoinf.2023.102457.
Lawlor, Jake, Francis Banville, Norma-Rocio Forero-Muñoz, Katherine Hébert, Juan Andrés Martínez-Lanfranco, Pierre Rogy, and A. Andrew M. MacDonald. 2022. "Ten Simple Rules for Teaching Yourself R." *PLOS Computational Biology* 18 (9): e1010372. https://doi.org/10.1371/journal.pcbi.1010372.
LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–44. https://doi.org/10.1038/nature14539.
LeCun, Yann, Yoshua Bengio, and others. 1995. "Convolutional Networks for Images, Speech, and Time Series." *The Handbook of Brain Theory and Neural Networks* 3361 (10): 1995.
Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." https://www.tensorflow.org/.
Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In, 80248035. Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
Ruan, Wenda, Keyi Wu, Qingchun Chen, and Chengyun Zhang. 2022. "ResNet-Based Bio-Acoustics Presence Detection Technology of Hainan Gibbon Calls." *Applied Acoustics* 198: 108939. https://doi.org/10.1016/j.apacoust.2022.108939.
Ruff, Zachary J., Damon B. Lesmeister, Cara L. Appel, and Christopher M. Sullivan. 2021. "Workflow and Convolutional Neural Network for Automated Identification of Animal Sounds." *Ecological Indicators* 124 (May): 107419. https://doi.org/10.1016/j.ecolind.2021.107419.
Scavetta, Rick J, and Boyan Angelov. 2021. *Python and r for the Modern Data Scientist*. O'Reilly Media, Inc.
Silva, Bruno, Frederico Mestre, Sílvia Barreiro, Pedro J Alves, and José M Herrera. 2022. "soundClass: An Automatic Sound Classification Tool for Biodiversity Monitoring Using Machine Learning." *Methods in Ecology and Evolution*.
Simonyan, Karen, and Andrew Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv Preprint arXiv:1409.1556*.
Sing, T., O. Sander, N. Beerenwinkel, and T. Lengauer. 2005. "ROCR: Visualizing Classifier Performance in r." *Bioinformatics* 21 (20): 7881. http://rocr.bioinf.mpi-sb.mpg.de.
Stevens, Eli, Luca Antiga, and Thomas Viehmann. 2020. *Deep Learning with PyTorch*. Simon; Schuster.
Stowell, Dan. 2022. "Computational Bioacoustics with Deep Learning: A Review and Roadmap." *PeerJ* 10 (March): e13152. https://doi.org/10.7717/peerj.13152.
Sugai, Larissa Sayuri Moreira, Thiago Sanna Freire Silva, José Wagner Ribeiro, and Diego Llusia. 2019. "Terrestrial Passive Acoustic Monitoring: Review and Perspectives." *BioScience* 69 (1): 1525. https://doi.org/10.1093/biosci/biy147.
Takhirov, Zafar. 2021. "Quantized Transfer Learning Tutorial." https://pytorch.org/tutorials/intermediate/quantized_transfer_learning_tutorial.html.
Ushey, Kevin, J. J. Allaire, and Yuan Tang. 2022. *Reticulate: Interface to 'Python'*.
Vu, Thinh Tien, Dai Viet Phan, Thai Son Le, and D. J. Clink. 2024. "Investigating Hunting in a Protected Area in Southeast Asia Using Passive Acoustic Monitoring with Mobile Smartphones and Transfer Learning." *Ecological Indicators*. 10.1016/j.ecolind.2024.112501.