Applied Acoustics 166 (2020) 107375

Contents lists available at ScienceDirect

Applied Acoustics

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

Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling

Ming Zhong [a,*], Jack LeBien [b], Marconi Campos-Cerqueira [b], Rahul Dodhia [a], Juan Lavista Ferres [a], Julian P. Velev [c], T. Mitchell Aide [b,d]

[a] AI for Good Research Lab, Microsoft, USA
[b] Sieve Analytics, San Juan, PR 00911, USA
[c] Department of Physics, University of Puerto Rico, San Juan, PR 00931, USA
[d] Department of Biology, University of Puerto Rico, San Juan, PR 00931, USA

ARTICLE INFO

Article history:
Received 24 December 2019
Received in revised form 6 March 2020
Accepted 9 April 2020

Keywords:
Deep learning
Convolutional Neural Networks (CNN)
Bioacoustic classification
Transfer learning
Pseudo-labeling

ABSTRACT

In this study, we evaluated deep convolutional neural networks for classifying the calls of 24 birds and amphibian species detected in ambient field recordings from the tropical mountains of Puerto Rico. Training data were collected using a template-based detection algorithm followed by a manual validation process. As preparing sufficient training data is a major challenge for many deep learning applications, we propose a novel approach that combines transfer learning of a pre-trained deep convolutional neural network (CNN) model and a semi-supervised pseudo-labeling method with a custom loss function to meet this challenge. Our proposed methodology enables the network to be trained in a supervised fashion with labeled and unlabeled data simultaneously, which effectively increases the size of training set and thus boosts the model performance. In classifying a test set of manually validated positive and negative template-based detections, our proposed model achieves 97.7% sensitivity (true positive rate), 96.4% specificity (true negative rate) and 99.5% Area Under a Curve (AUC). This multi-label multi-species classification methodology and its framework can be easily adopted by other acoustic classification problems.

1. Introduction

Camera traps [1–3], audio recorders [4,5], GPS tracking devices [6,7], and environmental DNA (eDNA) [8,9] have improved our ability to collect field data on fauna around the world. Although these technologies can produce terabytes of data, a major challenge is to convert the raw data into useful information for understanding species abundance and distribution. While animal sound detection and classification has recently attracted a wide range of interests, raw environmental audio recordings are particularly challenging for identifying individual species because a single one-minute recording can include many species of birds, frogs, mammals, and insects, along with abiotic (e.g. wind, rain) and anthropogenic (e.g. car noise) sounds.

To perform automatic species detection based on audio recordings, researchers have created species-specific algorithms by applying sound recognition techniques such as Gaussian mixture model (GMM) [10] or hidden Markov model (HMM) [11] where mel spectrum or mel frequency cepstral coefficient (MFCC) are usually used as input features. These approaches have been approved successful in identifying individual species, but they have limitations, especially in high diversity tropical ecosystems with hundreds of species where some species have very similar and overlapping calls. To address this challenge, much effort has been placed on developing effective methods to classify multiple species for acoustic data. Using the multi-instance multi-label (MIML) framework, a bag generator algorithm was proposed to convert an audio recording into a bag-of-instances representation, and a MIML classifier was applied to predict the set of species present in the recording [12]; an unsupervised feature learning method, spherical k-means, was used to generate feature representations classified using a random forest classifier [13].

Lately, deep learning has attracted much interest due to its capability to train huge amount of data and supremacy in terms of accuracy. Deep convolutional neural network (CNN) architectures have demonstrated great potential in classification problems as well as other tasks, such as object detection and image segmen-

* Corresponding author at: AI for Good Research Lab, Microsoft, Redmond, WA 98052, USA.
E-mail address: mizhong@microsoft.com (M. Zhong).

https://doi.org/10.1016/j.apacoust.2020.107375
0003-682X/

tation. Some famously known CNN architectures include AlexNet [14], VGG16 [15], and ResNet [16], among others. These models can successfully extract complex features from images and differentiate a high number of potentially similar classes, and have recently gathered popularity in the field of bioacoustics. For example, a convolutional neural network model was successfully implemented with MFCCs as input of the binary classification task for anuran sound [17]. In another collaborative data challenge, the best performing machine learning methods were convolutional and/or recurrent neural nets (CNNs, RNNs, or CRNNs) [18].

To achieve good performance, training a deep learning model typically requires large amount of data. However, using experts to obtain large number of labeled samples in acoustics is an expensive and time-consuming endeavor. In practice it may also be very difficult to collect enough labeled data, especially if a species rarely calls or if a species is rare. Given this scenario, transfer learning with fine-tuning [19] is a useful technique when the labeled dataset is relatively small. With transfer learning, a model trained on one task (or domain) is re-purposed on another related task (or domain). This approach is effective because the source model is usually trained on vast amounts of images, and certain low-level features learned from those images, such as edges, shapes and corners, can be shared across tasks. This can often improve performance and reduce the necessary training data and time, compared to training from an initially randomized state.

In addition to labeled data, incorporating unlabeled data into the training data can also improve model performance. Pseudo labeling [20] is a semi-supervised learning technique that uses a small set of labeled data along with a large amount of unlabeled data. With pseudo labeling, it initially trains a model on labeled data, and predicts labels on unlabeled data. It combines both datasets (i.e., data with true labels and data with predicted labels) as new training data, and then retrains the model. This approach provides a simple yet effective method to augment the size of training data, and incorporates the information contained in the unlabeled data during model training.

In this study, we compared the performance of three convolutional neural network models. The first model was trained using detected calls with positive labels only, and assumed all other species were absent. For this model we used the VGG16 architecture. The second model was similar to the first, but it was fine-tuned using the pre-trained ResNet50 weights. The third model used a novel approach that combined transfer learning and pseudo-labeling as a data augmentation technique to fine-tune a pre-trained CNN model, which resulted in an increase in model performance.

2. Approaches

2.1. Data collection and pre-processing

In this study, audio data was collected from about 700 sampling sites across the mountains of Puerto Rico from 2016 to 2019. Audio recordings were collected using portable acoustic recorders (Audiomoth and LG Android), which were programmed to record at a sampling rate of 48 kHz with 24 kHz bandwidth, using a medium gain (30.6 dB) and following a schedule of 1-minute audio recording every 10 min (i.e., in total 6 min of audio recordings per hour), 24 h per day for 1–2 weeks per sampling site.

To create the training and test data, we used the call template matching process for each species (see Table 1) within the ARBIMON II platform [21]. This process takes the time–frequency bounding-box coordinates of an example target call in a recording as input for each species (i.e., call template). The call template is then applied to a playlist of recordings to search for matching calls surpassing a chosen correlation threshold. All detections above the threshold and no more than the three closest matches per 1-minute recording were displayed to the user. The user annotated each detection as either positive or negative, indicating the presence or absence of the target species within the audio segment. The main focus of this study was species of Greatest Conservation Need according to the State Wildlife Action Plans [22] (see Table 1), but we also included some other common and abundant species in Puerto Rico (e.g., *Eleutherodactylus coqui*, *Margarops fuscatus*, *Turdus plumbeus*, *Patagioenas squamosa*). Among all the available labeled data from positive and negative detections, we randomly split them into training and testing portions, according to a 70/30 ratio; and then further split the training data into training and validation using the same ratio. As a result, there are 49% of

Table 1
Number of labeled data (including both present and absent detections) for each species, where "*" represents the species of Great Conservation Need.

Species Name	Taxon	Positive Detections	Negative Detections
Dwarf coqui (*Eleutherodactylus unicolor*) *	Frog	21,352	2981
Grass coqui (*Eleutherodactylus brittoni*) *	Frog	10,780	8157
Melodius coqui (*Eleutherodactylus wightmanae*) *	Frog	8582	13,976
Common coqui (*Eleutherodactylus coqui*)	Frog	7551	2293
Hedrick's coqui (*Eleutherodactylus hedricki*) *	Frog	5286	14,124
Cricket coqui (*Eleutherodactylus gryllus*) *	Frog	4371	11,793
Bronze coqui (*Eleutherodactylus richmondi*) *	Frog	4015	13,582
Locust coqui (*Eleutherodactylus locustus*) *	Frog	2839	3345
Red-eyed coqui (*Eleutherodactylus antillensis*)	Frog	1772	11,966
Upland coqui (*Eleutherodactylus portoricensis*) *	Frog	3487	5637
Gunther's white-lipped frog (*Leptodactylus albilabris*)	Frog	954	8932
Black-whiskered vireo (*Vireo altiloquus*) *	Bird	8984	13,351
Puerto Rican bullfinch (*Loxigilla portoricensis*) *	Bird	3728	18,815
Scaly-naped pigeon (*Patagioenas squamosa*)	Bird	2884	3132
Puerto Rican spindalis (*Spindalis portoricensis*) *	Bird	2297	11,886
Puerto Rican tanager (*Nesospingus speculiferus*) *	Bird	1929	10,539
Puerto Rican screech owl (*Megascops nudipes*)	Bird	1839	4039
Pearly-eyed thrasher (*Margarops fuscatus*)	Bird	1742	11,067
Elfin woods warbler (*Setophaga angelae*) *	Bird	1474	12,164
Bananaquit (*Coereba flaveola*)	Bird	1146	2372
Red-legged thrush (*Turdus plumbeus*)	Bird	1061	10,057
Puerto Rican woodpecker (*Melanerpes portoricensis*) *	Bird	1026	32,031
Puerto Rican tody (*Todus mexicanus*)	Bird	969	8386
Puerto Rican lizard cuckoo (*Coccyzus vieilloti*) *	Bird	481	8615

the data for training, 21% for validation, and the remaining 30% for out-of-sample testing.

Our manually validated dataset consists of 100,000 positive and 243,000 negative single-species detections across 24 species, where the duration of most of the detected calls was less than two seconds. The number of validated detections per species varied greatly because some of the species are rare or call infrequently (Table 1). To create CNN training samples from the call detections, mel spectrogram images were computed using 2-second audio clips from the start time of each detection. The *librosa* Python package [23] was used for mel spectrogram generation with default settings (sampling rate = 48 kHz, NFFT = 2048, hop length = 512, window length = 2048, Hann window). The resulting mel spectrograms had 24 kHz frequency bandwidth. Mel spectrograms were then converted to units of decibel (dB), resized to 224 by 224 pixels with RGB channels and stored as color images (see Fig. 1 for example of mel spectrogram from each species). The visualization of the mel spectrograms shows that the species calls can overlap in time and frequency, which makes the classification task more challenging. The color mel spectrograms were the input for the machine learning model and the corresponding single-species labels for each image (i.e. species present (positive) or absent (negative)) were used as the ground truth data for training and evaluating the multi-label multi-class classification model.

2.2. Model 1 – CNN using VGG16 architecture

In the first CNN model we used the neural network model VGG16 to classify the calls of the 24 species. The VGG16 architecture begins with the RGB images (size $224 \times 224 \times 3$) as input, and passes through five blocks of convolutional layers followed by three fully-connected layers. A rectified linear unit (ReLU) activation is performed right after each convolution and a max pooling operation is used at the end of each block. Two fully connected layers with 4096 ReLU activated units are then used before the final soft-max layer. When training the model, we used the Adam optimizer algorithm and an initial learning rate of 1e-4 with decay factor of 1e-7.

2.3. Model 2 – Transfer learning and fine-tuning with pre-trained CNN model

In the second CNN model we used the pre-trained ResNet50 weights with transfer learning to classify the calls of the 24 species. Transfer learning makes use of the knowledge gained while solving one problem and applying it to a different but related problem. Transfer learning enables us to leverage previous learnings when solving a different problem and start with weights already configured to detect various basic image features.

In our case, the model weights were initially trained on the ImageNet [24] dataset with 1000 classes of objects, but their pre-trained weights can be leveraged by a different task or domain [25]. With fine-tuning, we freezed some layers from the pre-trained model and only trained the last several layers. In this study, we used the pre-trained weights of ResNet50 and fine-tuned the parameters by adding a fully connected layer, a dropout layer and an output layer.

2.4. Model 3 – Transfer learning with custom loss function and pseudo labeling

In the third CNN model we used the pre-trained ResNet50 weights with custom loss function and pseudo labeling to classify the calls of the 24 species. With the CNN models described in the sections above, we only make use of the detected calls associated with positive labels (i.e., 100,000 clips), by assuming the absence of all other species and encoding them with negative labels. However, even within the 2-second time frame which the mel spectrogram was based on, there may be calls from multiple species. Therefore, the assumption of only a single species in a clip may not always hold and will yield some incorrect labels. Furthermore, the data of detected calls with negative labels (i.e., 243,000 clips) has not been used, which may include information that can be used for the classification task.

While in practice it is a challenge to collect and prepare large labeled datasets, pseudo labeling can be used to overcome this challenge. In this scenario, the initially trained model, which is based on labeled data only, will predict species presence or

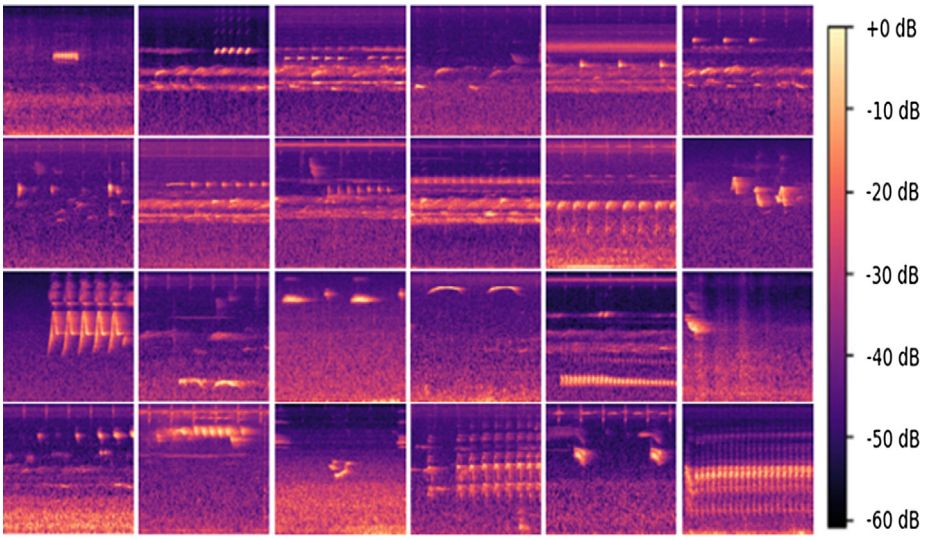

Fig. 1. Sample mel spectrograms for each species' two-second call. The horizontal axis represents time, and the vertical axis represents frequency (ranging from 0 to 24,000 Hz) with log transformation. Species names in the first row from left to right: *Eleutherodactylus unicolor*, *Eleutherodactylus brittoni*, *Eleutherodactylus wightmanae*, *Eleutherodactylus coqui*, *Eleutherodactylus hedricki*, *Eleutherodactylus gryllus*; second row from left to right: *Eleutherodactylus richmondi*, *Eleutherodactylus locustus*, *Eleutherodactylus antillensis*, *Eleutherodactylus portoricensis*, *Leptodactylus albilabris*, *Vireo altiloquus*; third row from left to right: *Loxigilla portoricensis*, *Patagioenas squamosa*, *Spindalis portoricensis*, *Nesospingus speculiferus*, *Megascops nudipes*, *Margarops fuscatus*; fourth row from left to right: *Setophaga angelae*, *Coereba flaveola*, *Turdus plumbeus*, *Melanerpes portoricensis*, *Todus mexicanus*, *Coccyzus vieilloti*.

absence for an unlabeled sample if the corresponding predicted probability surpasses a certain threshold. Then the model is re-trained in a supervised fashion with both labeled and unlabeled data simultaneously, where pseudo labels are assigned as if they were true labels.

With the techniques described above, we propose the following methodology for generating pseudo-labels when fine-tuning a pre-trained convolutional neural network model:

Step 1: Use one-hot encoding for the positive labeled data that is used for training, and represent the one species with positive label as 1 while all the remaining species as "Unknown".

Step 2: Use one-hot encoding for the negative labeled data that is used for training, and represent the one species with negative label as 0 while all the remaining species as "Unknown".

Step 3: Fine-tune a pre-trained convolutional neural network model (in this study we used ResNet50) with binary cross-entropy loss function only on data with labels either 1 or 0 (that is, do not penalize the predictions on the data with label "Unknown"). After n epochs (in our experiment, we chose a relatively small number $n = 5$ to avoid overfitting), stop training and make predictions for data with label "Unknown".

$$Loss = \begin{cases} -(y\log(p) + (1-y)\log(1-p)), & y \in \{0,1\} \\ 0, & y = \text{Unknown} \end{cases}$$

Step 4: Assign pseudo-label 1 to training data if the corresponding predicted probability is greater than a certain threshold (for example, 0.9), and assign pseudo-label 0 if the corresponding predicted probability is smaller than a certain threshold (for example, 0.1). All the other training data are assigned with label "Unknown" (that is, we only want to assign labels to data if the model has high confidence). Re-train the model with new labels with the loss function defined in Step 3.

To illustrate the steps described for model 3, the diagram of generating pseudo-labels is shown in Fig. 2.

3. Results

We used a default neutral threshold score of 0.5 for classifying the test dataset which includes 30,000 positive detections (with label "1") and 194,000 negative detections (with label "0"), and report the three key metrics: sensitivity, specificity and Area Under The Curve (AUC), of each model (see Table 2). Sensitivity, or true positive rate, measures the proportion of presence that was correctly predicted (i.e., TP/(TP + FN)); specificity, or true negative rate, measures the proportion of absence that was correctly predicted (i.e., TN/(TN + FP)). While sensitivity and specificity are dependent on the choice of threshold score, AUC provides an aggregate measure of performance across all possible classification thresholds (see Fig. 3 for ROC Curve for each model). Even though this is a multi-label multi-class classification model, for each detection we only evaluated the classification result of the single species that was detected by the template matching process and was manually validated with label, and we did not evaluate the remaining 23 species as their presence or absence are not annotated even though their pseudo-labels were generated by the model.

The specificity (true negative rate) was similar among the three models (the specificity can be increased by lowering the threshold score, or vice versa), but there were large differences in the sensitivity (true positive rate). When classifications were based on the positive labeled data and assuming the labels for all the other species as 0 (i.e. model 2), there was a 2% increase in sensitivity by fine-tuning the pre-trained CNN model (i.e. model 1). In contrast, as the size of negative labeled data is much larger than the positive labeled data in our dataset, incorporating it into our training by defining a custom loss function and generating pseudo-labels for

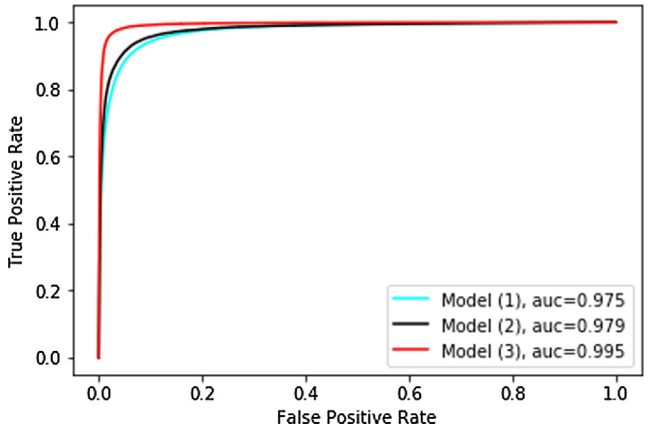

Fig. 3. The ROC Curve for each model. The proposed model (Pre-trained ResNet50 + Custom Loss Function + Pseudo Labeling) has the highest AUC value among three models.

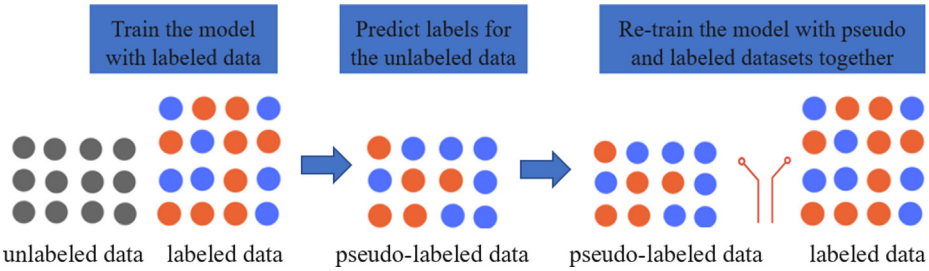

Fig. 2. The diagram of the proposed methodology with pseudo-label generating and model training. The dots with different colors represent observations with different labels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Classification results (sensitivity, specificity and AUC) by each CNN model.

Model ID	Model Description	Sensitivity (%)	Specificity (%)	AUC (%)
1	CNN with VGG16 Architecture	82.1	96.9	97.5
2	Pre-trained ResNet50	84.1	97.7	97.9
3	Pre-trained ResNet50 + Custom Loss Function + Pseudo Labeling	**97.7**	**96.4**	**99.5**

the unknown labeled data (i.e. model 3) yielded a significant increase in sensitivity (13.6%) over the results of model 2.

The histograms of predicted probability scores for the best performed model (model 3) show high confidence for scoring on the test dataset (that is, with predicted score >0.9 for positive labeled data, and <0.1 for negative labeled data, see Fig. 4). When implementing the model, the classification threshold scores can be adjusted depending on the importance of reducing false positives or false negatives. For example, thresholds closer to 1 would only predict species presence when there is a high level of confidence.

At the species level, model performance was strongly associated with the quantity of labeled data. We have conducted experiments of limiting different sample sizes of data within each species when training the model. The result shows that when the training sam-

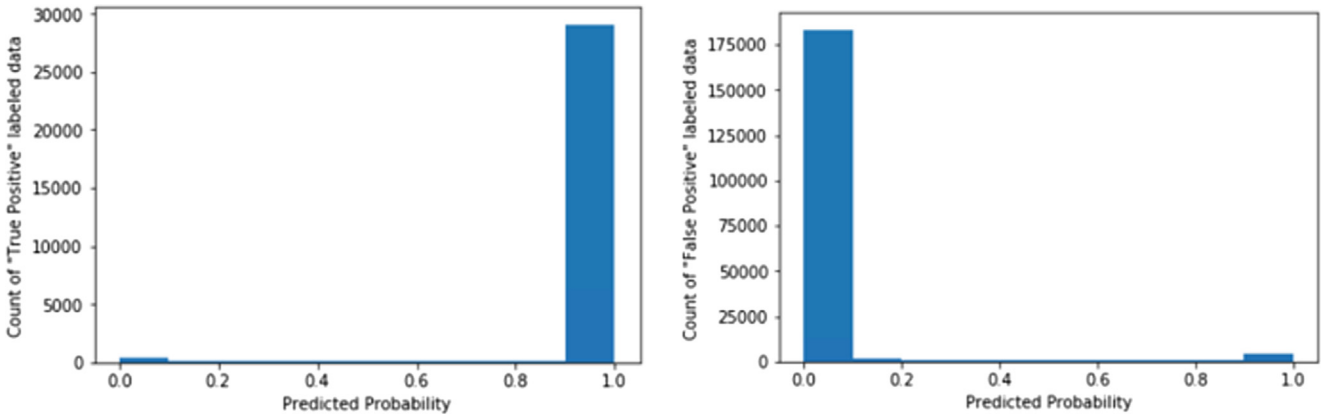

Fig. 4. Histograms of predicted probabilities from model 3. Left: Positive labeled test dataset consisting of 30,000 observations, with 97.7% of the observations had a predicted probably greater than 0.5. Right: Negative labeled test dataset consisting of 194,000 observations, with 96.4% of the observations had a predicted probably smaller than 0.5.

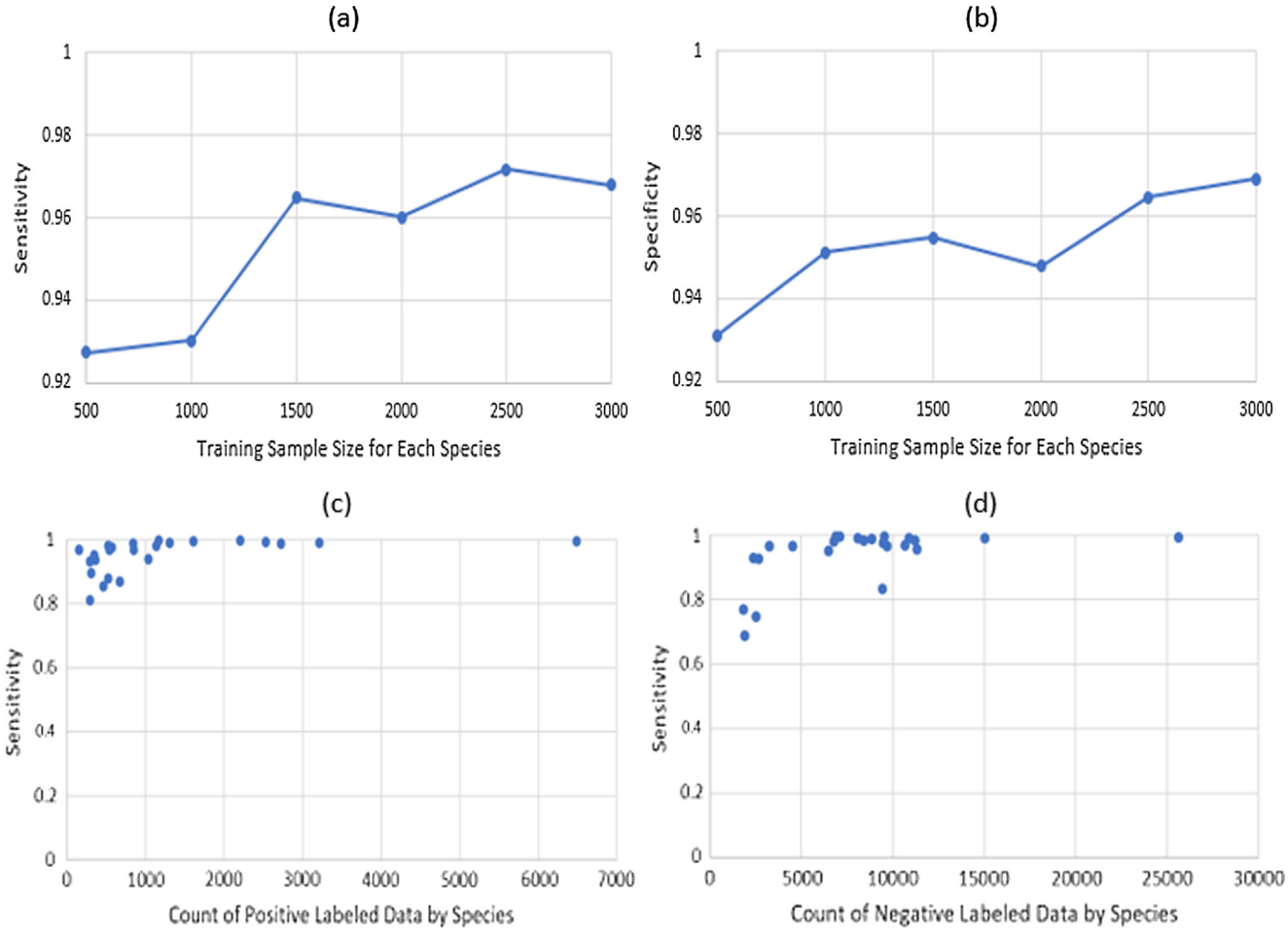

Fig. 5. (a): Model sensitivity for different sample sizes in the training dataset. (b): Model specificity for different sample sizes in the training dataset. (c): MODEL sensitivity for positive labeled species in the test dataset. (d): Model specificity for negative labeled species in the test dataset.

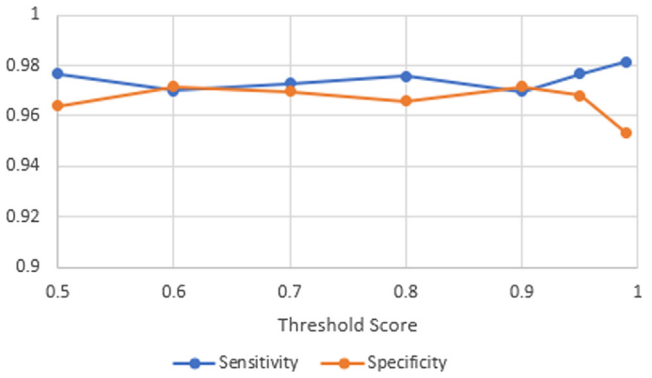

Fig. 6. The model sensitivity and specificity on testing dataset with different threshold scores when generating pseudo labels.

ple size of each species increased from 500 to 3000, the model performance (sensitivity and specificity) increases (Fig. 5(a) and (b)). Similarly, in the test dataset, there is a clear trend showing that the model has higher classification accuracies on the species with larger dataset sizes (Fig. 5(c) and (d)). This result partly explained why the inclusion of negative labeled data and the use of pseudo labels improve the performance of the classification model.

By including both positive and negative labeled data in the training process with pseudo labels, we were able to augment the effective size of the training dataset, which improved the model performance significantly. We conducted experiments by changing the threshold scores, and found that 0.9 (i.e. generate pseudo label "1" when predicted scores >0.9, and generate pseudo label "0" when predicted scores <0.1) is a good choice in this case (Fig. 6). This coincides with the result shown in Fig. 4, that for most of the observations, the predicted scores are either greater than 0.9 or smaller than 0.1. In general, when generating pseudo labels, it is suggested to only include observations that the initial model scores with high confidence (that is, scores either very high or very low), to avoid the model learning from incorrect labels.

4. Discussion

In this study, we demonstrate how transfer learning and pseudo labeling with deep convolutional neural networks (CNN) can achieve high sensitivity (>95%) and specificity (>95%) for the classification of calls from multiple birds and amphibian species from a tropical region. We provide both methodological and practical contributions by testing the performance of a machine learning approach to enable multiple bioacoustics classifications. While the CNN improves the template-matching performance, our proposed method also has the benefit of allowing CNN to train using template-based detection data, or in general, validated positive and negative detections from any detectors. Because the ground truth data was predefined by what the template matching process was able to identify, the annotated data used in this study might not be completely representative of the diversity of species calls present in the datasets. However, this approach enables that, if the recognition performance of the detectors is satisfactory in terms of not missing any calls (i.e., false negatives), or if the training data has been annotated from full audio recordings without detectors, classifications can be done on unseen full audio recordings.

With partially labeled data (that is, for each detection, only one species' presence or absence is known, while the remaining 23 species' presence or absence are unknown), the model 1 performs reasonably well (82.1% sensitivity and 96.9% specificity). Nevertheless, a pre-trained model and fine-tuning parameters from model 1 improves the performance of the classifications. Transfer learning enables the possibility of training deep convolutional neural network models with comparatively little training data. This is especially valuable in situations where large amount of labeled data is difficult to collect.

Although the best performing model (model 3) does not perform equally well among these 24 species, there is a clear trend showing the positive relationship between the size of training data and classification accuracy (Fig. 5). To further improve the model performance, some additional data augmentation methods, such as adding or removing background noises, cropping the original mel spectrograms, generating more mel spectrograms with time shift from the raw audios, or more advanced techniques such generating synthetic data using Generative Adversarial Networks (GANs) [26], may be helpful. Besides, in transfer learning, instead of using pre-trained models which are based on ImageNet, an alternative possibility is to pre-train a model from sound-only dataset.

Finally, the methodology and implementation framework presented in this study can easily be adopted for other bioacoustics problems, including classification of multiple species in a single model.

CRediT authorship contribution statement

Ming Zhong: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft. **Jack LeBien:** Data curation, Software, Validation, Writing - review & editing. **Marconi Campos-Cerqueira:** Data curation, Writing - review & editing. **Rahul Dodhia:** Writing - review & editing. **Juan Lavista Ferres:** Writing - review & editing. **Julian P. Velev:** Data curation, Writing - review & editing. **T. Mitchell Aide:** Data curation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank everybody who participated in the experiment for their support. This work was supported by AI for Earth grants at Microsoft. Our appreciation to Dan Morris for connecting different parties for fruitful discussions and helpful online materials.

References

[1] Ridout M, Linkie M. Estimating overlap of daily activity patterns from camera trap data. J Agri Biol Environ Stat 2009;14:322–37.
[2] Ahumada JA, Silva CEF, Gajapersad K, Hallam C, Hurtado J, Martin E, et al. Community structure and diversity of tropical forest mammals: data from a global camera trap network. Philos Trans Royal Soc B 2011;366:2703–11.
[3] Norouzzadeh MS, Nguyen A, Kosmala M, Swanson A, Palmer MS, Packer C, et al. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. PNAS 2018;115(25):E5716–25.
[4] Whytock RC, Christie J. Solo: an open source, customizable and inexpensive audio recorder for bioacoustic research. Methods Ecol Evol 2017;8:308–12.
[5] Hill AP, Prince P, Piña Covarrubias E, Doncaster CP, Snaddon JL, Rogers A. AudioMoth: evaluation of a smart open acoustic device for monitoring biodiversity and the environment. Methods Ecol Evol 2018;9:1199–211.
[6] Schofield G, Bishop CM, MacLean G, Brown P, Baker M, Katselidis KA, et al. Novel GPS tracking of sea turtles as a tool for conservation management. J Exp Mar Biol Ecol 2007;347:58–68.
[7] Bouten W, Baaij EW, Shamoun-Baranes J, Camphuysen KCJ. A flexible GPS tracking system for studying bird behaviour at multiple scales. J Ornithol 2012;154:571–80.

[8] Bohmann K, Evans A, Gilbert MTP, Carvalho GR, Creer S, Knapp M, et al. Environmental DNA for wildlife biology and biodiversity monitoring. Trends Ecol Evol 2014;29:358–67.
[9] Thomsen PF, Willerslev E. Environmental DNA - an emerging tool in conservation for monitoring past and present biodiversity. Biol Conserv 2015;183:4–18.
[10] Jančovič P, Köküer M. Automatic detection and recognition of tonal bird sounds in noisy environments. EURASIP J Adv Signal Process 2011;2011:1–10.
[11] Potamitis I, Ntalampiras S, Jahn O, Riede K. Automatic bird sound detection in long real-field recordings: applications and tools. Appl Acoust 2014;80:1–9.
[12] Briggs F, Lakshminarayanan B, Neal L, Fern XZ, Raich R. Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach. J Acoust Soc Am 2012;131:4640–50.
[13] Stowell D, Plumbley MD. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. PeerJ 2014;2:e488.
[14] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. NIPS, 2012.
[15] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ICLR, 2015.
[16] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: CVPR. p. 770–8.
[17] Fazekas B, Schindler A, Lidy T, Rauber A. A multi-modal deep neural network approach to bird-song identification. LifeCLEF working notes, 2017
[18] Stowell D, Wood MD, Pamula H, Sylianou Y, Glotin H. Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge. Methods Ecol Evol 2019;10:368–80.
[19] Shin H-C, Roth HR, Gao M, Lu L, Xu Z, Nogues I, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 2016;35:1285–98.
[20] Lee D-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. ICML workshop on challenges in 2013 representation learning, 2013.
[21] Aide TM, Corrada-Bravo C, Campos-Cerqueira M, Milan C, Vega G, Alvarez R. Real-time bioacoustics monitoring and automated species identification. PeerJ 2013;1:e103.
[22] Benson A. A national look at Species of Greatest Conservation Need as reported in State Wildlife Action Plans. U.S. Geological Survey Core Science Analytics, Synthesis, and Library, available: www.usgs.gov/core_science_systems/csas/swap/sgcn, 2016.
[23] McFee B, Raffel C, Liang D, Ellis DP, McVicar M, Battenberg E, Nieto O. librosa: audio and music signal analysis in python. Proceedings of the 14th python in science conference, 2015.
[24] Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F. ImageNet: a large-scale hierarchical image database. CVPR, 2009.
[25] Huh M, Agrawal P, Efros AA. What makes imagenet good for transfer learning? arXiv:1608.08614. 2016.
[26] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. NIPS, 2014.