Classification of Rare Hawaiian Birds through Audio Soundscapes. BirdCLEF 2022 Competition

ENSC 813
Final Project Report

Zhong Jia Xue
[redacted]

Introduction

To support the efforts of wildlife conservation, especially relating to avian species, passive acoustic monitoring (PAM) is essential to record data regarding the location and population of wildlife [1]. Birds in particular are important because they are found in most environments and can be used as an important metric to monitor eco-system health overall [2]. In the past, human observers would conduct studies at specific areas of interest, both visually and acoustically, to track bird counts within a given time period. However, manually tracking each encounter requires avian expertise and is time consuming. Thus, autonomous recording units (ARUs) are used to continuously monitor the points of interest. However, the problem remains that ARUs can easily generate tens of hundreds of TBs of data [1], which can result in much of the data being left unanalyzed. Novel machine learning techniques show promise in providing new tools for ecologists to survey biodiversity in sensitive ecosystems.

Therefore, the BirdCLEF 2022 competition is hosted on Kaggle to advance the state-of-the-art in machine learning approaches for analyzing lengthy audio recordings. This year, the competition is hosted by the Cornell Lab of Ornithology and is specifically targeted towards rare and endangered birds in Hawai'i, in the hopes of training robust classifiers which can support researchers in their effort to protect endangered Hawaiian birds [3]. The BirdCLEF 2022 Kaggle competition is the focus of this final project, in the hopes of adding value by developing and experimenting with techniques to reliably detect bird calls.

Fig. 1: Endangered Hawaiian Species Akiapolaau [4]

Dataset and Features

The dataset will come from the hosted Kaggle competition and consists of 32kHz sampled audio for 152 different bird species. This data was in turn collected from the Xeno-Canto foundation which hosts a repository of bird sounds from across the world. For each audio sample, meta-data is provided for the entire sample which includes categories like the primary bird species label, background bird species label, the type of bird call, location of recording and audio quality rating.

There are four main challenges with regards to the labels in this dataset. The first is that there may be background species present in the audio sample with no associated label, leading to noisy labels where the model may train on the wrong label. The second is that each bird species may have a few different audio signatures of different types, examples of these types would be “call”, “song” and “courtship call”. For the same bird species, these types of calls may be drastically different, requiring the model relate these sounds to the same bird.

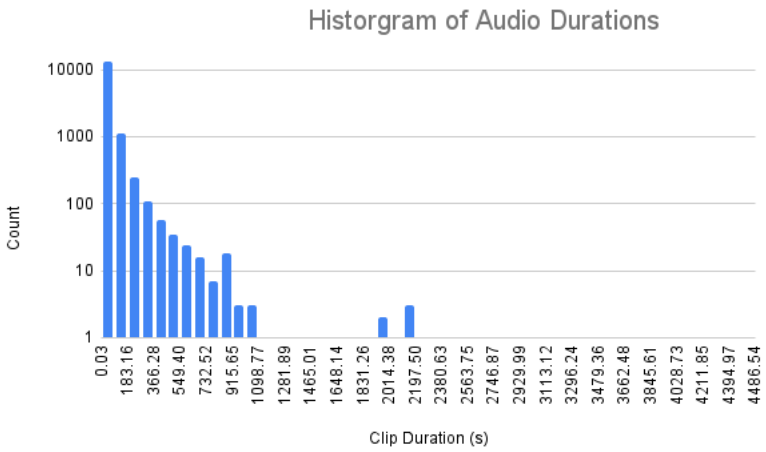

Fig. 2: Histogram Audio Clip Durations

The third challenge is that the length of audio clips is not uniform. As shown in Fig. 2 above, the duration of audio clips varies drastically. Some extreme examples would be the blkfra/XC649198.ogg and the Commyn/XC548866.ogg audio files, the former audio clip having only 0.03 seconds of audio and the latter audio clip having a lengthy 1 hour and 14 minutes of audio. In Fig. 2, we can see that the majority of clips have a duration of less than 1000s, with a few samples with greater time. For each species, the total length of training audio varies widely, with some species only having a few minutes of training data, to other species having hours. This challenge relates to a broader issue common in the machine learning community, which is the lack of high-quality annotated data.

The fourth challenge is that there may be large periods of silence hidden within the training audio. This will pose a problem when segmenting audio clips for the model to train on because the model may train incorrectly on data samples of full silence. Shown below in Fig. 3, we can see that this audio clip with the label of Black-crowned Night Heron contains a full 22 seconds of silence, which will mislead the model during training.

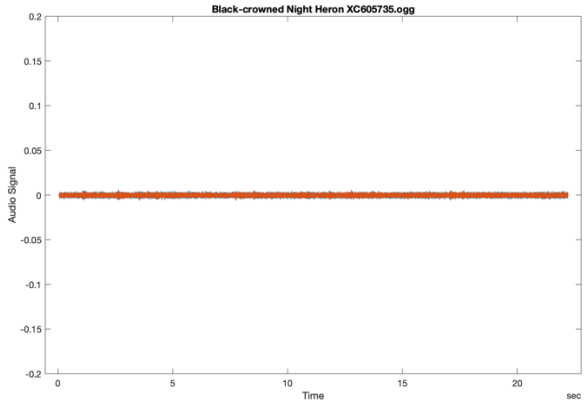

Fig. 3: Black-crowned Night Heron Silent Clip

Scoring Method

To calculate the competition score, a hidden dataset of 5500 recordings of approximately 1 minute long will be populated and the model must output True/False for whether a bird's call is present in any 5 second segment. However, only 21 out of the 152 birds present in the training dataset will be scored in this competition. The public leaderboard is only calculated on 16% of the test data, whereas the rest of the data is saved for the private leaderboard. The evaluation metric is kept hidden, however the competition organizers reveal that the macro F1 score most similarly resembles the metric used. The F1 score is a harmonic mean of precision and recall, and the equation is show below [5].

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

Literature Review

Due to the nature of the BirdCLEF competition, the results and methodologies of previous years participants are available to current participants to improve on. In 2021, many participants chose to use deep convolutional neural networks after converting the input audio data to a mel scale spectrogram [1]. Many participants opted to use available CNN architectures such as ResNet [6] or EfficientNet[7] with weights trained on the ImageNet dataset. Majority of teams also applied the Mixup and Specaugment data augmentation techniques during training.

The 1st place team in the competition last year, Team **Dr.北村の愉快な仲間たち** [8], employed three main stages to train their model.

The first is a nocall detection model, which is used to filter out audio samples with no birdcalls present. The team used this nocall detector to weigh the labels of segments, which the model predicts no birdcalls are present, for training the next model. Since the data given by the BirdCLEF2021 competition is insufficient for training the nocall model, the team used the data from the freefield1010 archive [9], which provides fine-grained call / no-call labels for 10 second audio clips. The no-call model used the ResNeXt50 architecture.

In the second stage, the team used the BirdCLEF2021 competition data and converted the audio into a mel spectrogram before training, with 128 mel frequencies in the frequency range of 0Hz to 16000Hz. Then, using ResNeSt50[10] as the model, the team multiplied the bird call predictions from stage 1 with the annotated label for that audio segment and used that to train the model. Any audio segments with secondary labels then have their labels further multiplied by 0.6 since it is assumed that birds with secondary labels are much less likely to appear than birds with primary labels.

For the third stage, the results of the second stage, and additional metadata like date, location and if the bird call was predicted by the model in the previous or following audio segments, were used to train LightGBM for binary call / nocall classification.

The 6th place team in the competition last year, Team **Just do it** [11], also focused on a strong nocall detection system. The team converted all input audio files into mel-scale spectrograms with 128 frequency channels. The team also used ResNeSt50[10] as the model backbone. One of the main differentiators of this team is that they manually labelled regions with long no-call sections, as opposed to labelling each individual bird call, which would be time consuming. The team also noticed that authors of audio clips hosted on xenocanto would cut audio at the beginning or end of clips if no birdcall was present. Thus, the team assumed that the first and last 5 seconds of each clip contained a bird call.

The 2nd place team in the 2020 competition, Team **NPU-BAI** [12], had a traditional approach in building their pipeline, basing the model on the Xception neural network, and processing the inputs with spectrogram conversion. The most notable contribution is the use of the mix-up data augmentation technique where data samples are overlayed upon each other. Their choice of architectures is due to previous BirdCLEF competitions where Inception based models are proven to perform well on Mel spectrograms.

Although there may be a variety of methods for performing deep learning audio classification, the go-to method for current and previous BirdCLEF competitions have been CNN based methods on spectrogram processed inputs. The hosts of the 2021 competition have acknowledged that very few participants have explored approaches with 1D convolutional networks or with transformers [1]. From the three participants mentioned above, all three top performing methods used spectrogram inputs as well as audio data-augmentation techniques. Team **Dr.北村の愉快な仲間たち** and team **Just do it** had a strong focus on no-call detection. The first placed team utilized a ResNeXt50 model to train an automated no-call classifier, whereas the sixth place team used dataset heuristics and manual labelling to identify no-call audio segments. As with the current BirdCLEF2022 competition, a strong no-call classifier will continue to be a key component. The main issue with the manual labelling from the 6th place team is that the process is very time consuming, and thus for this final project, the automated labeling of no-call is a more suitable option.

Methodology

As shown in the literature review above, the go-to tool for this competition is the mel-spectrogram with a CNN based model. A no-call detector is also critically important. One of the main values added will be that the backbone model from previous years will be developed in Tensorflow instead of Pytorch, whereas the majority of competition notebooks are developed in PyTorch. The second piece of value added will be the addition of hidden layers and output layers on top of available architectures to suit the multi-label classification problem of BirdCLEF2022. These added layers will also include additional regularization techniques like dropout and norm penalties. The third piece of value added will be the training of multiple model architectures and ensembling these models for a higher model accuracy.

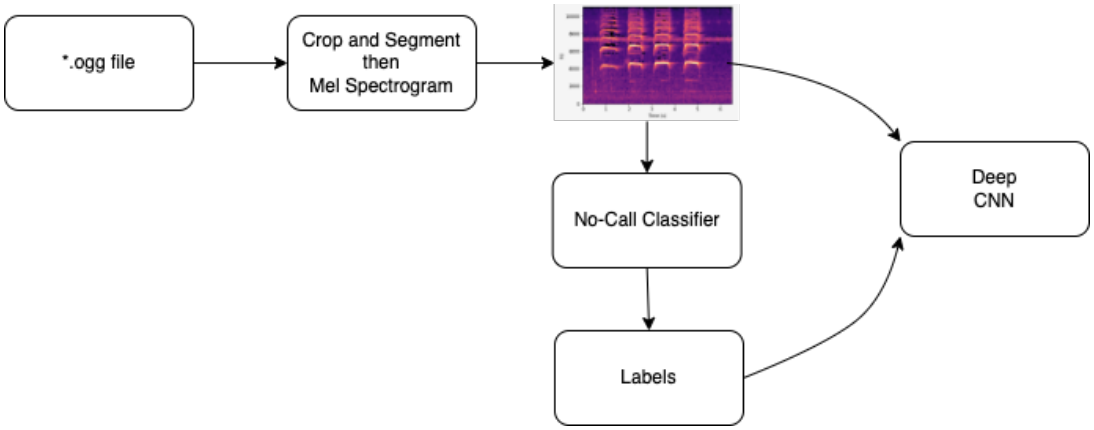

Fig. 4: Proposed Training Pipeline

Shown above is the proposed training pipeline. The input audio file is first cropped into 7 second segments starting every 5 seconds. This is to ensure that any birdcall on the edge of the crop will be included in the audio segment, following that of the first-place team last year [8]. Audio clips shorter than 7 seconds are padded by repeating the available audio, instead of the conventional zero-padding. This is to prevent the model from training on black mel-spectrogram outputs (zero padded) when a portion of cropped audio may still contain a birdcall.

The mel-spectrogram configuration is set for 128 mel-scaled frequencies with an f_{min} = 0kHz and f_{max} = 16kHz. Following the first-place team, the output image would have a resolution of 128x281. The image was then normalized into 32-bit floating point and stacked 3-fold, imitating RGB, to allow the use of transfer learning for selected model architectures.

The ResNeXt50 model from [8] was retrained on the same freefield1010 archive data to adjust labels weights for training the main CNN model. We can see the validation score for training on the right, showing signs of overfitting after the 4th epoch. This is similar to the recorded score of 0.89 in [8].

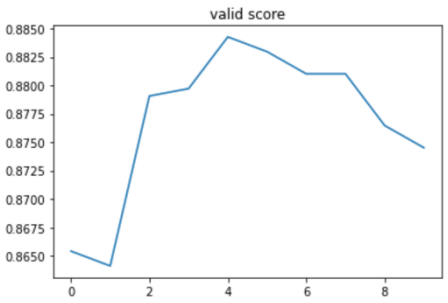

Fig. 5 No-call Model Training

For training the main model, ResNet50V2 and Xception models are chosen due to their success in previous competitions, especially since Inception-based models perform well on Mel-spectrogram inputs [12]. EfficientNet was also chosen as another off-the-shelve model to experiment with.

```
global_average_pooling2d (Glob  (None, 2048)        0           ['post_relu[0][0]']
alAveragePooling2D)

dense (Dense)                   (None, 1024)        2098176     ['global_average_pooling2d[0][0]'
                                                                ]

dropout (Dropout)               (None, 1024)        0           ['dense[0][0]']

dense_1 (Dense)                 (None, 152)         155800      ['dropout[0][0]']

==================================================================================================
Total params: 25,818,776
Trainable params: 25,773,336
Non-trainable params: 45,440
__________________________________________________________________________________________________
```

Fig. 6 Added Top Layers to ResNet50V2 Architecture

As seen in Fig. 6, the following fully connected layers are added to the chosen networks to fit the BirdCLEF2022 competition.

- GlobalAveragePooling was chosen over flatten to reduce the number of trainable params when connecting to the 1024 neuron hidden layer
- An L2 norm penalty was added in the hidden layer to help with regularization
- Dropout of 0.5 was also added in the hidden layer
- An output layer with 152 neurons fitting the number of Bird Species
- Sigmoid activation is used for the output layer, fitting a multilabel classifier
- Binary cross-entropy is chosen as the loss function
- Network will transfer learn from ImageNet weights

Resulting from the addition of these output layers, the final ResNet50V2 model will contain around 25 million trainable parameters.

Training and Local Results

In terms of training, the majority of training was completed locally on a Nvidia 2070 Super, with a small subset of training completed on Google Colab. A batch size of 64 is manually tuned to prevent the local machine from running out of memory. The training will be conducted over 35 epochs and 33% of training samples will be reserved for the validation set. The Adam optimizer is used with an initial learning rate of 0.001. In addition, early stopping and model check pointing is used, both monitoring the validation loss. The learning rate is also dynamically reduced when the validation loss is seen to plateau, due to an observation that models benefit if the learning rate is decreased once the learning stagnates [13]. Mixup data augmentation is also applied to later training results [14].

An initial attempt at training was conducted with a modified pipeline. This pipeline introduced an additional output class of no_call to the main model, for a total number of 153 classes. The idea was to filter labels such that if the no_call classifier determined that a sample has no calls (with a 0.5 threshold), the training label would be changed to no_call instead of the primary bird species.

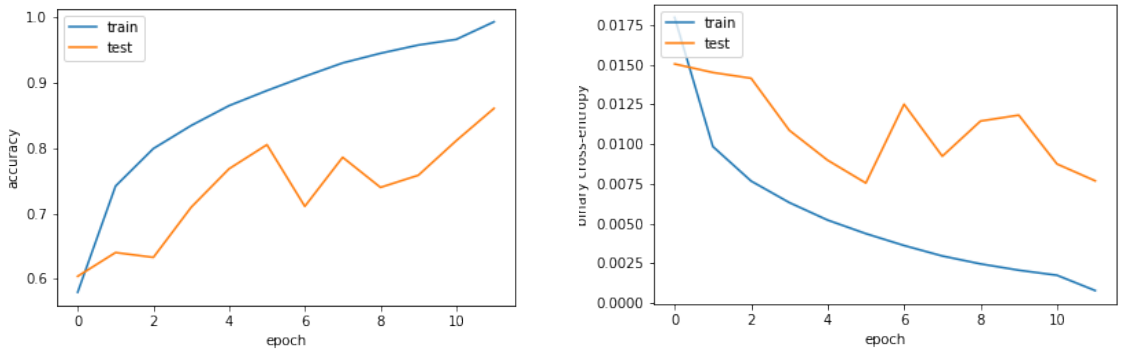

Fig. 6: ResNet50V2 Training with Explicit NoCall Label

From Fig. 6, we can see that the validation loss (right) fails to decrease with the training loss and the training is stopped at epoch 12 due to early stopping. One hypothesis as to why the model is performing poorly is that there may be sigmoid saturation since the labels are encoded to 0 and 0.99.

From the results of the initial attempt, the pipeline is changed to reflect the methodology described above where the training labels are multiplied by the no_call probabilities. Thus the issue of sigmoid saturation is resolved since the labels will be weighed by the no_call model's output.

ResNet50V2

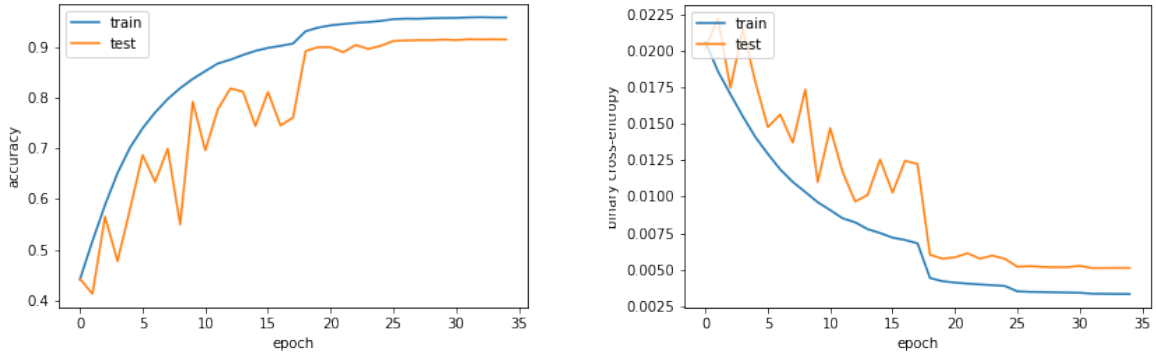

Fig. 7: ResNet50V2 Training with Weighted Labels

As shown in Fig. 7, the model accuracy (left) and model loss (right) for ResNet50V2 are portrayed after training for 35 epochs. From the graph, there is a major drop in model validation loss at epoch 18. It is hypothesized that this is caused by ReduceLROnPlateau reducing the learning rate from 0.001 to 0.0002 on epoch 18.

Xception

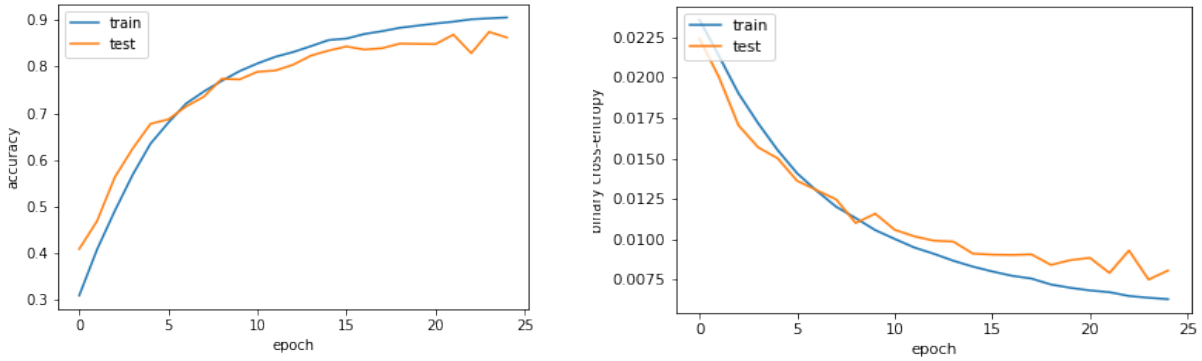

Fig. 8: Xception Training with Weighted Labels

Shown in Fig. 8, is the model accuracy (left) and model loss (right) for training Xception over 30 epochs. The validation curve is noticeably smoother compared to that of ResNet50V2.

EfficientNetB5

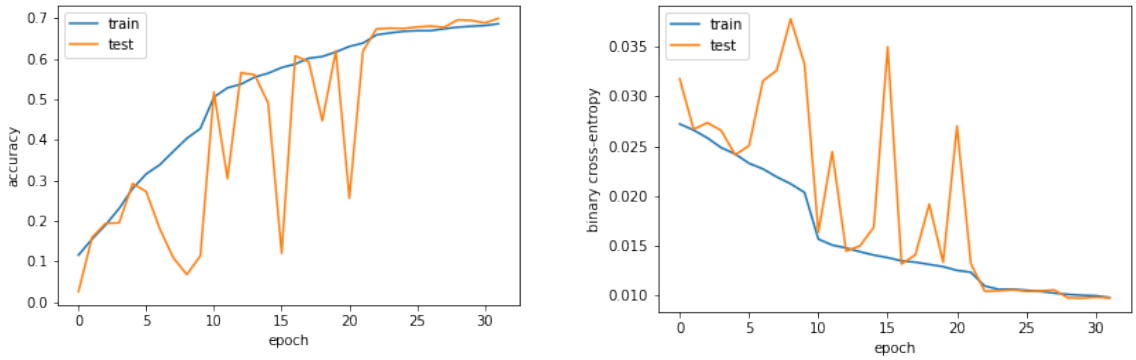

Fig. 9: EfficientNetB5 Training with Weighted Labels

As shown in Fig 9, EfficientNetB5 fits the data poorly, and has very large spikes on the validation data. The reason for the spikes in the validation curves may be due to a variety of factors such as large learning rate, noisy dataset samples or small batch size.

Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
ResNet50V2	0.9581	0.0033	0.9148	0.0051
Xception	0.9039	0.0064	0.8747	0.0075
EfficientNetB5	0.6896	0.0100	0.6950	0.0097

Table 1: Model Scores After Training

As we can see from the training results, ResNet50V2 provides the best performance, reaching a validation accuracy of 0.91 and validation loss of 0.0051. At a close second is Xception with a validation accuracy 4% lower than ResNet50V2. Lagging far behind is EfficientNetB5 with a validation accuracy at 0.6950.

Label: hawhaw Call Prob: 0.58
ResNet50V2 Predictions: {0.6968, ‘hawhaw’}, {0.0099, ‘blkfra’}

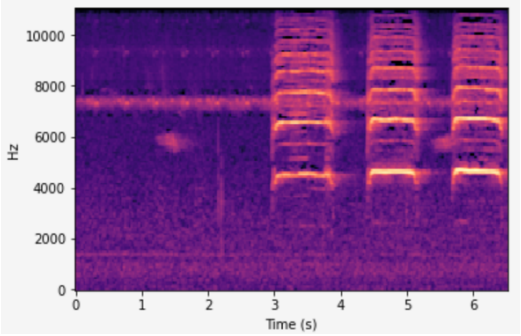

Fig. 10: Hawaiian Hawk Melspectrogram XC124707.ogg


Fig.11 Hawaiian Hawk [15]

A visual example is provided above for the local inferencing results of a Hawaiian hawk audio clip. The mel-spectrogram received a call probability of 0.58 and the ResNet50V2 model predicted a ‘hawhaw’ bird with a confidence of 0.70.

Kaggle Results and Discussion

As local validation results often differ from Kaggle results, the Kaggle Public leaderboard score is shown below.

Methodology	Public Leaderboard Score
ResNet50V2 with Explicit NoCall	0.49
ResNet50V2	0.52
ResNet50V2, Xception and EfficientNetB5 Ensemble	0.54
ResNet50V2 and Xception Ensemble	0.55
ResNet50V2 w/ Mixup	0.55
ResNet50V2 and Xception Ensemble w/ Mixup	0.57

Table 2: Kaggle Public Leaderboard Results

From the Kaggle competition results above, we can see that the lowest score is obtained from ResNet50V2 with the explicit no_call category. The next score is from the ResNet50V2 model with the no_call probability weighed labels. Then the ensembling of ResNet50V2, Xception and EfficientNetB5 contains the next highest score at 0.54. The ensemble without EfficientNetB5 has an even higher score, likely because EfficientNet performed poorly even in local validation thus bringing the Ensemble down. Mixup Data augmentation proves to significantly help the model generalize, as the ResNet model’s score increases by 0.03 with Mixup. Next, the ensemble with ResNet and Xception is the highest scoring combination.

From these results, we can clearly see that weighing the bird call labels with the no-call probability achieves a higher result than that of an explicit no-call category. Another clear result is that Mixup augmentation increased the public leaderboard score by 0.3 when only

ResNet50V2 was used. In addition, unsurprisingly, the ensembling of multiple models gives higher results than that of each model individually.

In terms of the Kaggle results, there is a large difference in the local validation scores and the Kaggle public leaderboard scores. This may be due to a few factors. There may be a large difference in the quality or noise level of audio between the training dataset and the test dataset. The training audio was scrapped from xeno-canto, a large online repository of birds and the test dataset is hidden. The test dataset is either also scraped from an online repository, or a private data source belonging to the competition hosts. The second reason may be that the leaderboard score only calculates 21 out of the 152 birds in the training data. This may be the largest contributing factor, since the local validation score is for all 152 birds.

Regardless of the factors mentioned above, the author acknowledges that the leaderboard score is not quite as high as the 0.65 leaderboard score that the author proposed in the project proposal. Any additional value added can be from the implementation of the training pipeline in Tensorflow, the experimentation with various off-the-shelve models and the results from ensembling such models.

Conclusion

The development of machine learning models for birdcall identification can help researchers monitor the health of a particular eco-system. Through an in-depth literature survey, the main methodologies for the audio classification of bird species is done through training convolutional neural networks on mel-spectrogram cropped audio samples. By using a separate no-call detection model, the training sample labels are then weighed by the no-call probabilities. Three models, ResNet50V2, Xception and EfficientNetB5, are trained and their respective results are discussed. The public leaderboard results for these models, and their ensembles are shown.

Future work can include further data augmentation techniques on the specific 21 bird species scored in the competition. Experiments into other model architectures and the tuning of hyperparameters for different ensembles can also be explored.

References

[1] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, and A. Joly, "Overview of BirdCLEF 2021: Bird call identification in soundscape recordings," *CEUR Workshop Proceedings*, vol. 2936, 2021.

[2] S. Kahl, M. Clapp, W. A. Hoppin, H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, and A. Joy, "Overview of BirdCLEF 2020: Bird Sound Recognition in Complex Acoustic Environments," *CEUR Workshop*, vol. 2696, 2020.

[3] S. Kahl, "Birdclef 2022," *Kaggle*, 2022. [Online]. Available: https://www.kaggle.com/competitions/birdclef-2022/overview. [Accessed: 09-Apr-2022].

[4] E. VanderWerf, "Akiapolaau - eBird," *ebird.org*, 2013. [Online]. Available: https://ebird.org/species/akiapo. [Accessed: 09-Apr-2022].

[5] K. Leung, "Micro, Macro & weighted averages of F1 score, clearly explained," *Medium*, 09-Jan-2022. [Online]. Available: https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f. [Accessed: 09-Apr-2022].

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

[7] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[8] N. Murakami, H. Tanaka, M. Nishimori, Birdcall Identification using CNN and Gradient Boosting Decision Trees with Weak and Noisy Supervision, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, (2021).

[9] D. Stowell, M. Plumbley, An open dataset for research on audio field recording archives: freefield1010, Journal of the Audio Engineering Society (2014).

[10] H.Zhang,C.Wu,Z.Zhang,Y.Zhu,H.Lin,Z.Zhang,Y.Sun,T.He,J.Mueller,R.Manmatha, et al., Resnest: Split-attention networks, arXiv preprint arXiv:2004.08955 (2020).

[11] M. Shugaev, N. Tanahashi, P. Dhingra, U. Patel, BirdCLEF 2021: Building a birdcall segmentation model based on weak labels, in: CLEF Working Notes 2021, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2021, Bucharest, Romania, (2021).

[12] Bai, J., Chen, C., Chen, J.: Xception based system for bird sound detection. In: CLEF working notes 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020).

[13] “Keras Documentation: Reducelronplateau,” Keras. [Online]. Available: https://keras.io/api/callbacks/reduce_lr_on_plateau/. [Accessed: 13-Apr-2022].

[14] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[15] “Hawaiian hawk,” Wikipedia, 02-Apr-2022. [Online]. Available: https://en.wikipedia.org/wiki/Hawaiian_hawk. [Accessed: 13-Apr-2022].