Jared Huzar, Philip Genovese, Craig Fox, Ben Rapp
CIS 4496
Projects in Data Science
11 April 2022

Model and Performance Report for Rainforest Species Audio Detection

Problem Introduction

One of the major issues that we face as a planet is the loss of biodiversity in our rainforests. Rainforests are responsible for a very high percentage of the biodiversity present on earth and maintaining a balanced, diverse ecosystem is very important to many facets of human health. Unfortunately, biodiversity is decreasing and many rare species require careful supervision in the rainforest. One of the best and easiest ways to monitor species' health is using audio data. Because many species are difficult to find visually, Rainforest Connection has created a product called RFCx, which automatically monitors audio on the forest floor to aid conservation management efforts. The presence or absence of rainforest species is an indicator of climate change and habitat loss. Therefore, being able to monitor rainforest species constantly is key to preserving rainforest health. Unfortunately, identifying species from audio recordings manually is a difficult, time-consuming task that requires highly skilled experts. Consequently, conservation efforts can be greatly benefitted from a machine learning model that can identify rainforest species from audio clips. To train a model, a collection of audio files with minimal labeling along with some metadata is provided regarding the song type.

The audio data is in FLAC files. The associated information is divided into two csv tables: one for true positives and one for false-positives. The false-positive data is significantly larger than the true-positive dataset. The true-positives were developed by experts identifying some species throughout the audio clip. The data collectors then developed a simple CNN prediction model and ran the data through it. The experts then reviewed the predictions and identified cases where it was incorrect. These incorrect marks make up the false-positive dataset, i.e. cases where a species is identified as not present in a clip. The interesting part of this data is that it is very incomplete. While we do not know the exact number of species present in a recording, on average only one species is identified at one point in the recording. From qualitative experience listening to the clips, there are at least a hundred calls in each recording. This presents a huge challenge. The other interesting fact is the correlation between the true positives and the false-positives. Even though it was a simple model, the false-positives are still cases that would be difficult to distinguish. This underscores the importance of the false-positive dataset. Incorporating it into the model means it will especially help the cases where our true positive model was most off.

Feature Engineering

Our feature engineering pipeline revolved around the overall process of audio representation learning. We have experimented with three different representations of audio clips. To generate all three representations initially, the raw audio data was read, sliced, and centered around the known bird calls. Information on the start and end time of the known bird call in the audio data was provided as part of the metadata. Each audio file was centered around the median of that time window and then was also sliced to remove data substantially outside of the time window containing the signal. After this step of processing the raw audio data, our three feature engineering methods diverge.

However, once each audio representation is generated, the data augmentation process for each is identical. Data augmentation is necessary for this challenge because of the very limited size of the training dataset and the weak labeling of the false-positive dataset. Data augmentation is used to provide more samples with ground truths for the model to train on, without requiring more expert labeling of audio files. For audio data, augmentation can occur either on the raw data or on the spectrograms themselves. Since the representations are what is being fed into the model, we chose to augment the representations themselves. For each of the chromagrams, mel-spectrograms, and spectrograms, we created a duplicate of them and added uniformly distributed background noise to all of the duplicate representations. This effectively, based on a markedly improved performance with data augmentation, mitigated the effects of two major obstacles inherent in our data. Firstly, this pipeline doubled the size of our training dataset as the model was now fed both the original representations and those with the background noise added. Secondly, it reduces the effect of the background noise which comes from actual noises in the rainforest. By adding this uniform noise to all the duplicated representations, the model is less likely to learn the background noise as it is not unique to any species.

Mel-Spectrograms

Our initial focus was on converting the audio data into mel-spectrograms. Mel-spectrograms are commonly used as input to machine learning models in other Kaggle notebooks publicly available for this task. Using a short-time Fourier transform, the audio data is converted from the time-series input into the frequency domain as a magnitude spectrogram and then converted to the mel-scale. Mel-spectrograms are designed to represent a signal’s amplitude, or loudness, over various frequencies.

Therefore, they are effective representations of bird calls which are variable in frequency and the associated amplitude. In addition to slicing the data based on the time interval, for the mel-spectrogram representation we also sliced based on the frequency interval. In the metadata, the max frequency of the birdsong is provided. To promote our model concentrating on learning this signal and not noise, we removed any data which is above the provided frequency. We also normalized the mel-spectrogram. Next, we attempted to denoise the actual signal by decomposing it using a nearest neighbor filter. This replaces each data point with the aggregate of its nearest neighbors in feature space. This is meant to remove or mitigate the effect of any random perturbations or disruptions to the observed birdsong signal. This mel-spectrogram representation was then duplicated and augmented with uniform noise as described previously.

Figure 1. Mel Spectrogram Representation with and without Augmentation

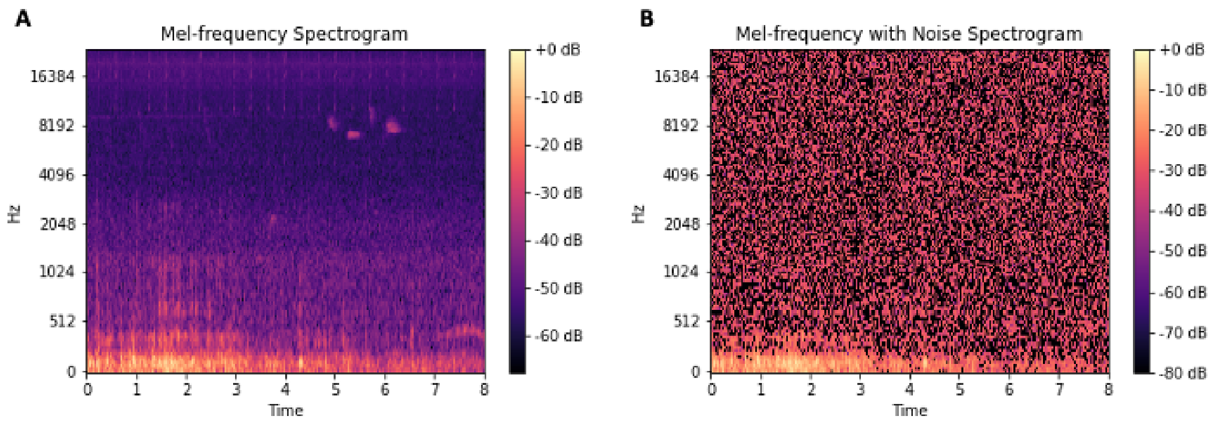

Fig 1. Example of mel-spectrogram representations from a bird call. **(a)** Mel-spectrogram without added background noise. **(b)** Mel-spectrogram with added background noise.

As evident in figure 1, the signal representing the birdsong, present in the bottom-left of the mel-spectrogram, is still retained after the background noise augmentation.

Chromagrams

To generate chromagrams from the time series data, the data is again converted into the frequency domain using the short-time Fourier transform. In the case of the spectrogram representation, a power spectrogram is computed. However, a chromagram is an energy spectrogram instead of a power one. The chromagram is constructed using the same sample rate as the spectrograms because the sample rate is a property intrinsic to the data collection.

Spectrograms

To generate the spectrograms, like the mel-spectrograms the time series data was converted to the frequency domain using the short-time Fourier transform into a magnitude spectrogram. Unlike the mel-spectrogram, the spectrogram is not converted to the mel scale and the magnitude spectrogram is its final representation. Both the spectrogram and the mel scale spectrogram visualize the strength of the signal over different frequencies. However, the mel-spectrogram is specifically converted to best represent the audio signals which humans can hear. For the case of the model which is performing image analysis, this change in scale should not have a significant effect on the classification performance.

Modeling Algorithms

We experimented with 3 different neural network architectures which are popular and high performing in image classification tasks. Based on background research, we found converting the audio classification into an image classification problem is the most common and effective approach. For each sample, all the models are fed a 224 x 224 x 3 array. This consists of the 224 x 224 image representation stacked 3 high. Initially, the target was simply the ID corresponding to the ground truth species. However, we have improved our model so that now the target corresponding to each sample is a vector of length 24, with each element of the vector corresponding to the probability of a species being in the given audio clip. For the ground truth species, their probability is set to 1. For the false-positive species, their probability is set to 0, and for all the other species their probability was set to 0.05. This was done so that we could incorporate our false-positive data into training. The false-positive dataset contains almost 4x the number of samples as the true positive dataset so incorporating it alongside the true positive dataset results in an almost 5x larger training set than if we solely used the true positives.

ResNet50

ResNet50 is an artificial neural network (ANN) initially developed by researchers at Microsoft. More specifically, ResNet50 is a residual neural network. The distinction is that residual networks use skip connections which allow the model to jump over layers. Adding these skip connections allows ResNet models to circumvent the vanishing gradient problem which is a hurdle in deep learning models. This works because the skip connections simplify the network by using fewer layers during the beginning of the training phase. These layers are then gradually restored as the data's feature space is learned. Also, the residual network architecture is able to mitigate the degradation problem which also plagues deep learning models. The degradation problem is when adding more layers causes the accuracy to saturate and then degrade. Deeper models consequently lead to higher training errors. The skip connections of ResNet mitigate this by using identity mapping. The skip connections connect the shallow layers to the deeper layers with simply an identify function, leaving the output of the previous layer unaltered as input to the deeper layer. ResNet's effectiveness in practice has been demonstrated through performance on the ImageNet database, and it won 1st place in the ImageNet Large Scale Visual Recognition Challenge 2015 classification task. For our application, we used ResNet50 pre-trained on millions of images from the ImageNet database.

To adapt the weights to our particular classification problem, we replaced the last ResNet50 layers and then trained using the provided training data and the processed audio signals. We replaced the last 3 layers with linear layers using Relu activation between them and the softmax activation function for the final output. In addition, for the 2 hidden layers, we applied a dropout rate of 0.2. This is a normalization method implemented to reduce the likelihood of overfitting. Our ResNet implementation uses stochastic gradient descent (SGD) with a momentum of 0.9. We chose to use momentum because Sutskever et al. showed that SGD with momentum and a slowly increasing scheduler can train deep neural networks to very high levels of performance. Consistent with this, we included a scheduler with step size of 7 and gamma of 0.4. For the loss function, we use BCE with Logits Loss. The BCE with Logits Loss is optimized for multi-class classification and it is commonly used in similar audio classification problems. We were inspired to use it in our model based on the Ren et al paper which identifies if an individual has COVID-19 based on their cough, and the Wu et al paper which proposes a new audio representation learning method. Both papers used BCE with Logits Loss. Our model is trained until convergence occurs, we found that 20 epochs are required.

ResNeSt

ResNeSt is the most recently developed method of the three and it is a split-attention neural network. ResNeSt is a variant of the aforementioned ResNet which instead stacks split-attention boxes. ResNest is unique in that it applies channel-wise attention on different network branches to learn diverse representations and capture interactions across features. ResNeSt outperformed EfficientNet in accuracy and latency trade-off in image classification. For our implementation, we loaded the ResNeSt pre-trained on the ImageNet database and then adapted it for our specific problem using the same procedure as described above for ResNet50. All model hyperparameters remained consistent from ResNet to ResNeSt

EfficientNet

EfficientNet, a newer architecture than ResNet but older than ResNeSt, is a convolutional neural network that introduces an advanced scaling method. Typically, when scaling neural networks the number of layers is increased, but it can also be done by increasing the width of the network, i.e, adding more nodes per layer. Specifically for image classification challenges, the input resolution of the image can also be increased. EfficientNet differentiates itself in that it uses a compound coefficient to scale up the CNN in a structured manner. Scaling up each dimension of a network increases model performance, but balancing the three dimensions results in optimal performance. The relationships between the three dimensions are first determined using a grid-search to determine the compound coefficient. With this knowledge of the compound coefficient, the model can then be scaled up appropriately balancing the three dimensions. This scaling method was shown to improve ResNet's ImageNet accuracy. Additionally, EfficientNet is a new baseline network that was developed by performing a neural architecture search. In our model, we implement EfficientNet-B7, the most complex and the most accurate EfficientNet variant. To train the model for our challenge specifically, one linear layer is added as is a 0.2 dropout rate. The hyperparameters and loss function again remained consistent with those used in the ResNet50 and ResNeSt implementations.

Metrics

The Kaggle evaluation metric is label-weighted label-average ranking precision (LWLARP). Label weighted average precision quantifies for each ground-truth the fraction of higher ranking labels that were true labels. The score ranges from 0 to 1. 0 is the case where all the higher-ranking labels than a ground truth label are not found in the sample. 1 is the case all the labels above a ground truth are also all found in the sample. Consequently, a higher score is indicative of better model performance. Since the metric is also label-weighted, the overall score is the average of all the labels in the test set. Each label receives equal weight. Consequential to our model, this means the exact probabilities are unimportant but the relative probabilities are significant. Also, since each label receives equal weight, our model can't disregard the species that have less ground truth observations as they are equally important in the overall scoring.

Our validation and training metric is accuracy. Since the probabilities of every species being in the audio clip are not part of the training data, it is not useful to use the LWLARP for validation. The model is considered correct if the most probable species is the ground truth label. Accuracy is a suitable metric because the ground truth label should be the most probable, which is what the accuracy measures, and this would maximize the LWLARP since the LWLARP is dependent on relative probabilities.

Evaluation

While the model is not yet in its final form, we have evaluated the model in its current iteration as well as all previous iterations. The relative performances across iterations have guided the model and feature engineering decisions we have made and the components of the pipeline which are implemented in the most current iteration.

Figure 2. Effect of Data Augmentation on Model Performance

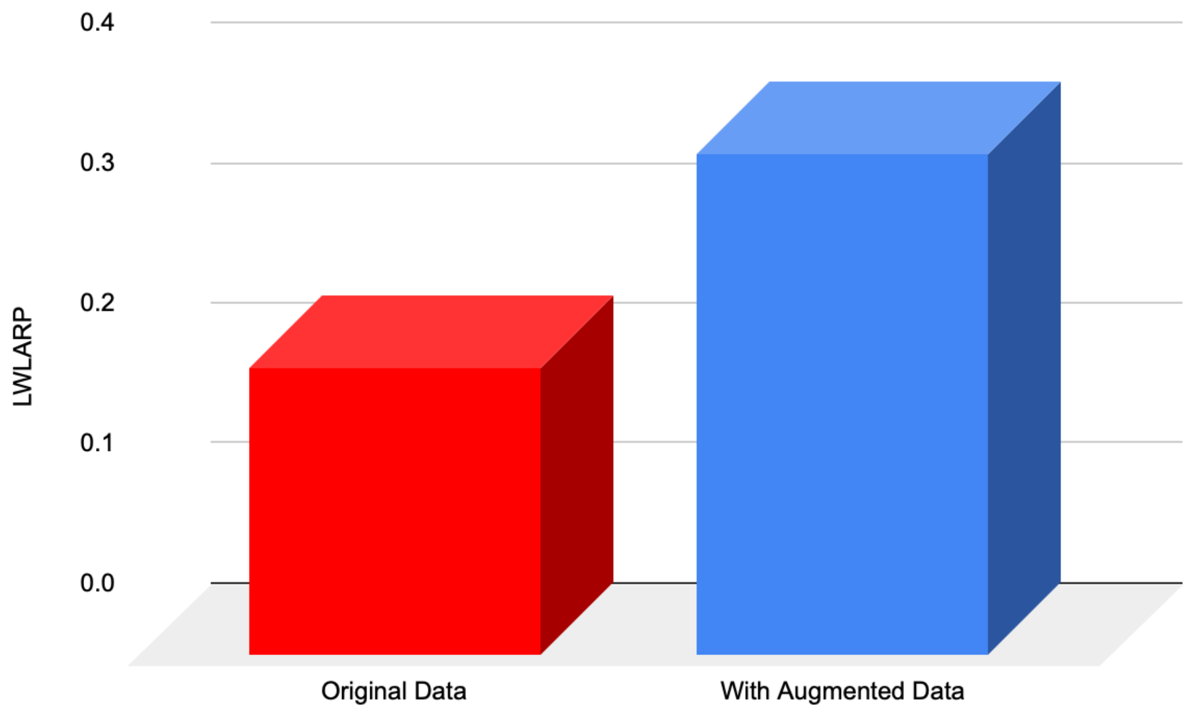

Figure 2. With the addition of augmented data, ResNet had significantly greater performance.

For example, we observed a significant improvement in our model's LWLARP from the inclusion of data augmented with uniform background noise, as seen in figure 2. This has subsequently been incorporated into the iterations of our models which we have since developed. We also evaluated the performance of the three model architectures mentioned previously. ResNeSt performed the best, while ResNet performed the worst.

Figure 3. Performance of Different Model Architectures

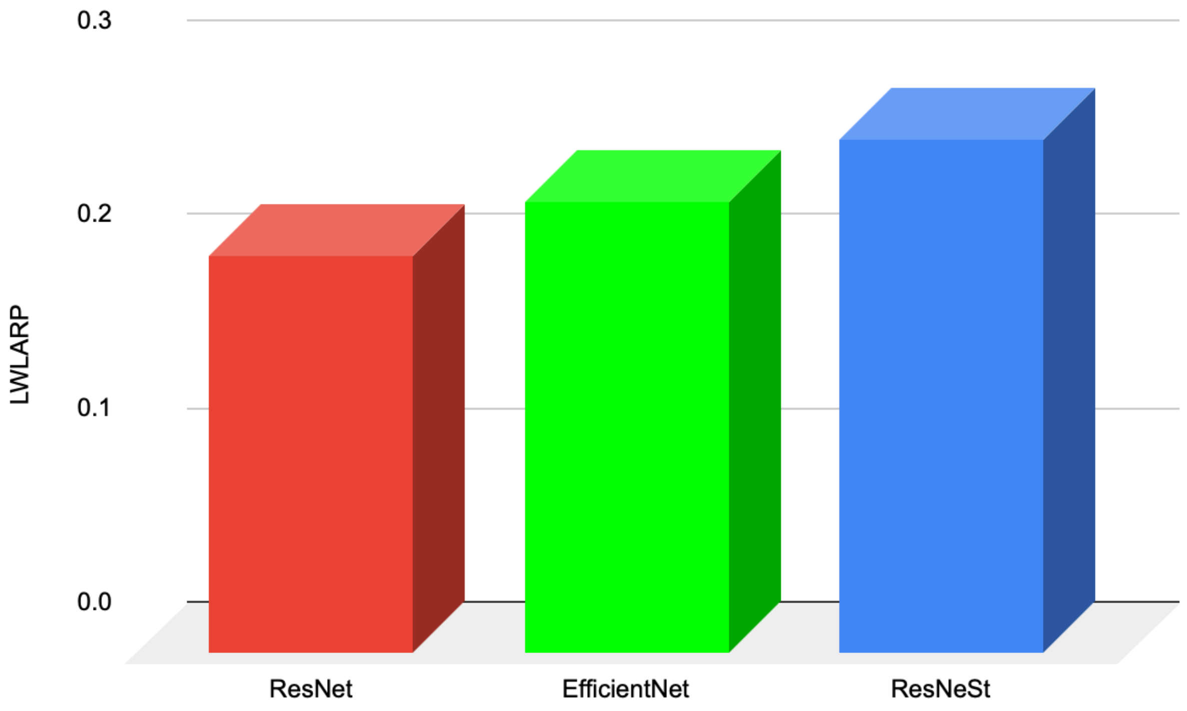

Figure 3. The performances of the three model architectures on only the true positive dataset represented as mel-spectrograms.

Another component of our pipeline that improved performance is the normalization of the mel-spectrogram along with the frequency masking. Implemented with ResNet50, this preprocessing step resulted in an almost 2x increase of our LWLARP to 0.58982 as seen in figure 4.

Figure 4. Effect of Frequency Masking and Spectrogram Normalization

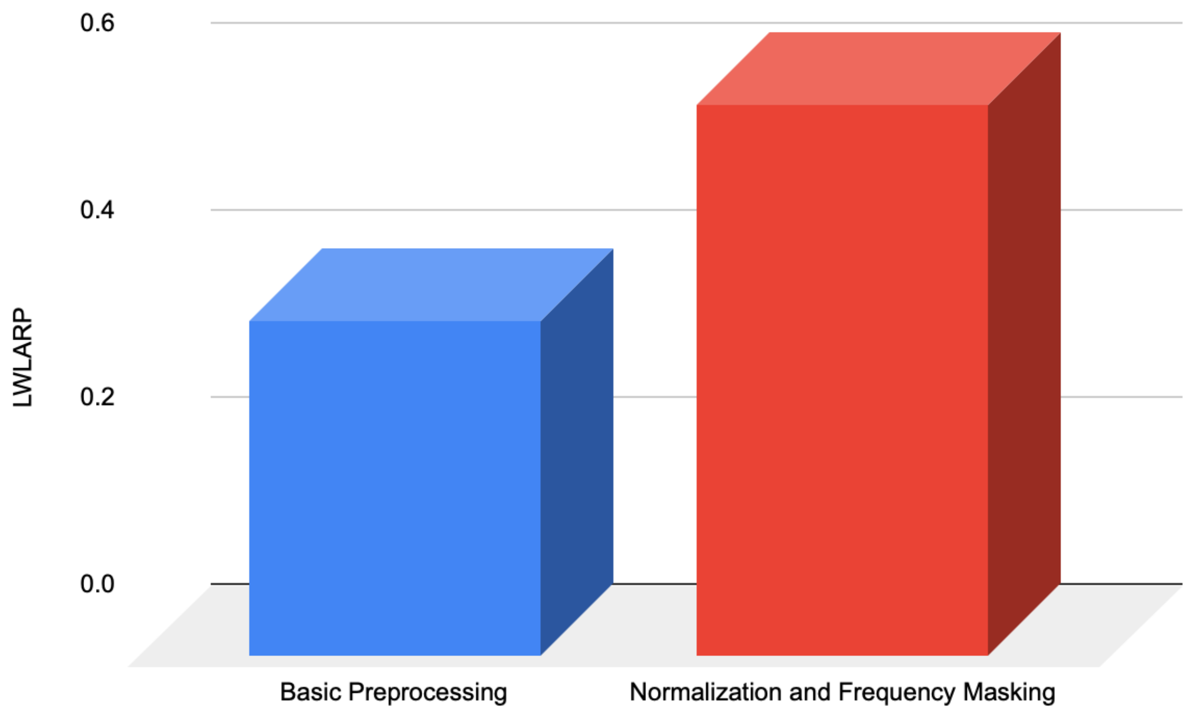

Figure 4. The effect of normalizing the spectrogram and masking it to the frequency range of the known birdsong. These results are from ResNet50 with true positives and augmented data.

Overall we have seen our model sequentially approve significantly over the past 2 months to almost triple its initial performance.

<u>Figure 5. Sequential Tracking of Challenge Performance</u>

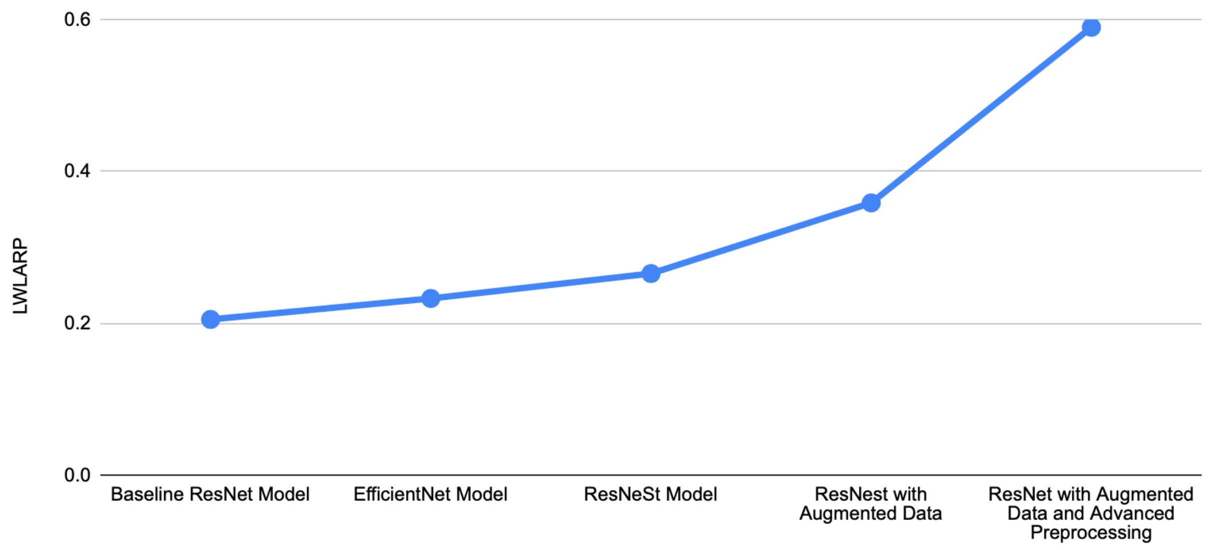

Figure 5. Our performance on the species detection classification problem beginning from our baseline model to our most recent iteration.

In addition to continually improving, an important aspect of our performance is the similarity between our training accuracy and our validation accuracy. This similarity indicates our model does generalize well and overfitting is not occurring.

References

BCE with Logits Loss Documentation:
https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

Brownlee, Jason. “A Gentle Introduction to Dropout for Regularizing Deep Neural Networks.” *Machine Learning Mastery*, 6 Aug. 2019, https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” *ArXiv*, Cornell University, 10 Dec. 2015, https://arxiv.org/pdf/1512.03385.pdf%3E.

“Rainforest Connection Species Audio Detection.” *Kaggle*, https://www.kaggle.com/competitions/rfcx-species-audio-detection.

Ren, Zhao, et al. “The EIHW-GLAM Deep Attentive Multi-Model Fusion System for Cough-Based COVID-19 Recognition in the DiCOVA 2021 Challenge.” *ArXiv.org*, 6 Aug. 2021, https://arxiv.org/pdf/2108.03041.pdf.

Sutskever, Ilya, et al. “On the Importance of Initialization and Momentum in Deep Learning.” *Computer Science University of Toronto*, 2013, https://www.cs.toronto.edu/~hinton/absps/momentum.pdf.

Tan, Mingxing, and Quoc V Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” *ArXiv.org*, 11 Sept. 2020, https://arxiv.org/pdf/1905.11946v5.pdf.

Wu, Ho-Hsiang, et al. “Wav2CLIP: Learning Robust Audio Representations from Clip.” *ArXiv.org*, 15 Feb. 2022, https://arxiv.org/abs/2110.11499.

Zhang, Hang, et al. “ResNeSt: Split-Attention Networks.” *ArXiv.org*, 30 Dec. 2020, https://arxiv.org/pdf/2004.08955.pdf.