ANIMAL SOUND CLASSIFICATION USING DEEP NETWORKS

ZHEN HUANG [ZTHUANG@SAS], NIMAY KUMAR [NIMAY512@SEAS], YINGCHUAN WANG [WANGY57@SEAS],

ABSTRACT. In this project we investigate using deep networks to identify animals from the sounds they make. Past work has investigated using Convolutional Neural Networks in order to classify animal sounds, to varying degrees of success. There are various methods for embedding the audio as input for the network, including wave signal, spectrogram, and Mel Frequency Cepstral Coefficient (MFCC). One difficulty of this task is the lack of a large, unified dataset. Prior work found that data augmentation and transfer learning greatly boosted performance with CNNs[6]. We aim to experiment with the audio embedding options and data augmentation to see how they affect performance. In addition, we would like to explore the use of Audio Spectrogram Transformers and Recurrent Neural Networks in classifying audio. Code is available: `https://github.com/NimayRKumar/in_the_jungle`

1. INTRODUCTION

Audio processing and recognition has become an important part of our increasingly digitized lives. Many of these tasks utilize deep networks, whether it be generation for voice assistants, or classification for speech recognition. One application of sound classification is identifying animals by the sound they make. Listening to soundscapes is a method researchers use to track biodiversity[3], which is especially important amidst the deforestation and loss of wildlife occurring today. In particular, bird calls are a good measure of a rainforest's well-being. However, collecting audio data and labeling them is a slow and expensive process, and the amount of available data for this task is currently limited. With this project, we hope to shed some light on what models perform well for audio classification on which data representations.

2. CONTRIBUTIONS

We tested out three different neural network architectures fitted to two different audio data representations, and experimented with data augmentations. The architectures we examined were CNNs, RNNs, and Audio Spectrogram Transformers (ASTs). Our data representations included spectrograms and Mel-frequency Cepstral Coefficients (MFCCs), and for augmentations we used Gaussian noise and air absorption. We found that our choice of data augmentations did not significantly improve the performance of our models, and in some instances they worsened. Overall our best results were achieved with a CNN trained on un-augmented MFCC data, followed closely by the performance of the AST (on spectrogram data). We trained an RNN with and without using LSTM, but these networks both had the worst performance of all and are not well-suited to audio classification tasks via spectrograms or MFCC representations.

3. BACKGROUND

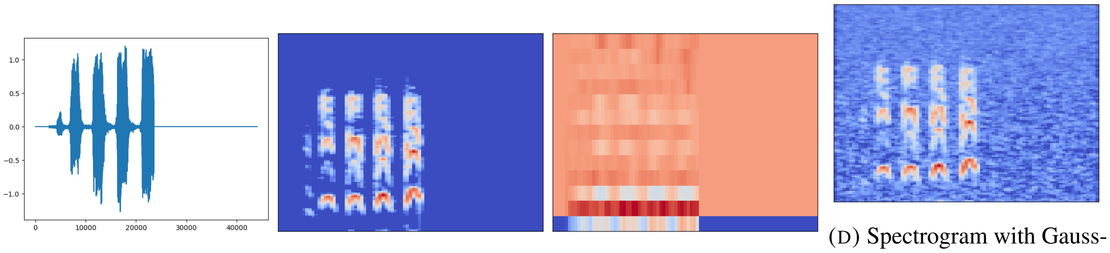
In sound classification, there exists a lack of substantial data for deep learning due to the cost of data collection. Furthermore, the available datasets are not well-balanced. Common animals such as dogs and cats have much more training data than ones such as donkeys or lions. Sound data also has a unique structure from text or images. For audio files, there are three common ways to extract features from them for learning.

3.1. Wave Signal.

An audio can be represented as an one dimensional array given a sample rate. The sample rate refers to the number of samples taken per second to represent the audio. The standard sample rate for audio CDs is 44.1 kHz, which corresponds to a sample rate of 44100. Nyquist's theorem states that a periodic signal, in this case sound waves, must be sampled at more than double the highest frequency of the signal. Therefore, for an audio file in 44.1 kHz, the highest frequency that can be captured is 22.05 kHz. For this reason, in Librosa and many other audio processing packages, the default sample rate is 22050.

3.2. Mel-Spectrogram.

Mel-Spectrograms are obtained by applying short time fourier transform (STFT) and Mel filter to the wave signals. This converts the signal from a time domain to a frequency domain, allowing us to examine the intensity for each frequency at a time stamp.



(A) Audio Signal (B) Spectrogram (C) MFCC (D) Spectrogram with Gaussian Noise

FIGURE 1. Features Extracted From A Monkey Audio

3.3. **Mel-frequency Cepstral Coefficients (MFCC).** MFCC is a compact representation of the spectral features of a sound signal through a number of coefficients. MFCC drastically reduces the dimension by extracting the important and discarding the less relevant information. The empirical convention is to use 13 coefficients, as the extraction maintains a good balance between compactness and robustness.

4. RELATED WORK

The work of Şaşmaz et al[1] explored using CNNs to classify animal sounds, and used the LibROSA Python library to process audio files into spectrograms and MFCCs. Gong et al[2] introduce the Audio Spectrogram Transformer, a convolution-free attention-based architecture for audio classification using spectrograms. Lastly, the work of Sun et al[4] investigated using CNNs trained on augmented spectrograms to classify animal sounds. We aim to build on these works by experimenting with combinations not yet explored. Using same data set as Şaşmaz et al[1], we train models on both spectrograms and MFCCs created used LibROSA. Furthermore, we train our models with both unaltered data and data augmented with Gaussian noise and air absorption.

5. APPROACH

5.1. **Data Preprocessing.** The dataset we used from Şaşmaz et al[1] consists 875 animal sounds from 10 types of animal. Specifically, the dataset contains audio from 200 cat, 200 dog, 200 bird, 75 cow, 45 lion, 40 sheep, 35 frog, 30 chicken, 25 donkey, and 25 monkey. We note that the data is heavily unbalanced, which we will discuss briefly later.The audio files are in the form of wav file, ranging from less than one seconds to more than five seconds. The audio contains minimum background noise. Using noise reduction to further lower the background noise will result in the animal sound being muffled, thus reducing the information within the data. As a result, no preprocessing is done to remove background noise. The animal sound can be heard within the first 1 second of the video. The audios are converted to wave signals via Librosa with a sampling rate of 22050. The signals are padded or trimmed such that they equate to 2 seconds of audio data. We used an 80/20 split of the signals to create our training and validation sets, respectively.

5.2. **Data Augmentation.** Because our core dataset has only 875 samples we artificially increased the size of our dataset by augmenting. Using the audiomentations Python library, we have seven augmentations available to us: AddGaussianNoise, AirAbsorption, BandPassFilter, GainTransition, RepeatPart, TimeStretch, and TanhDistortion. Each augmentation applied to the original training set of size n adds an additional n samples to our training set. Due to memory limits, in our experiments we only applied two augmentations: AddGaussianNoise and AirAbsorption. All the signals are converted to log mel spectrogram, which is on a decible scale, and Mel-frequency cepstral coefficients (MFCC). Figure 1 shows the raw signal, spectrogram, MFCC, and spectrogram augmented with Gaussian noise associated with the audio of a monkey's call.

We created the following models for each the spectrogram and MFCC data, with the exception of AST which is only suited for spectrograms

5.3. **Recurrent Neural Network (RNN) and LSTM.** We first experimented with RNNs, however they proved to be sub-optimal, perhaps because they fail to effectively capture spatial dependencies. When training on MFCC data, our RNN achieved a training accuracy of 75%, but the validation accuracy oscillates at 34%. The spectrogram training loss fails to converge and heavily oscillates, and this is reflected in the validation accuracy which never changes from 2.28%. The RNN likely overfits the MFCC training set and simply fails to learn on the spectrogram training set. We then implemented an image LSTM, whose structure is shown in Figure 2 below.

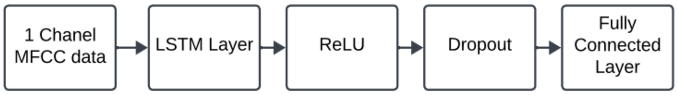


FIGURE 2. LSTM structure

The LSTM RNN achieves a validation accuracy of 79% on MFCC data, and 40% on spectrogram data. This model doesn't fail to converge and demonstrates promising results when trained on MFCC data.

5.4. **Convolutional Neural Network (CNN).** VGG16 is a well known architecture that consists of repeated blocks of convolutional layers with small filters, enabling effective feature extraction. We created two CNNs inspired by VGGs, one each for MFCC and spectrogram data, which differ only in their layer size. Figure 3 shows the model, which consists of several convolutional layers, batch normalization, and a ReLU, followed by a max-pooling layer for extracting the feature map. After convolution, the feature maps are flattened and passed through the fully connected layers for classification, in which we also use dropout for regularization. We use cross entropy loss and SGD for our optimizer, with a batch size of 35.

5.5. **Audio Spectrogram Transformer (AST).** The audio spectrogram transformer was developed by Yuan Gong, Yu-An Chung and James Glass in 2021 [2]. It is a convolution-free and purely attention based model for audio classification. We load in an AST model pretrained using ImageNet and AudioSet, and train using cross entropy loss and SGD.

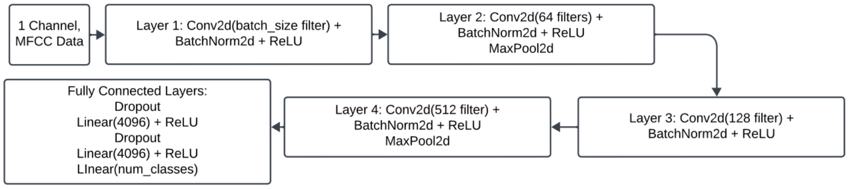


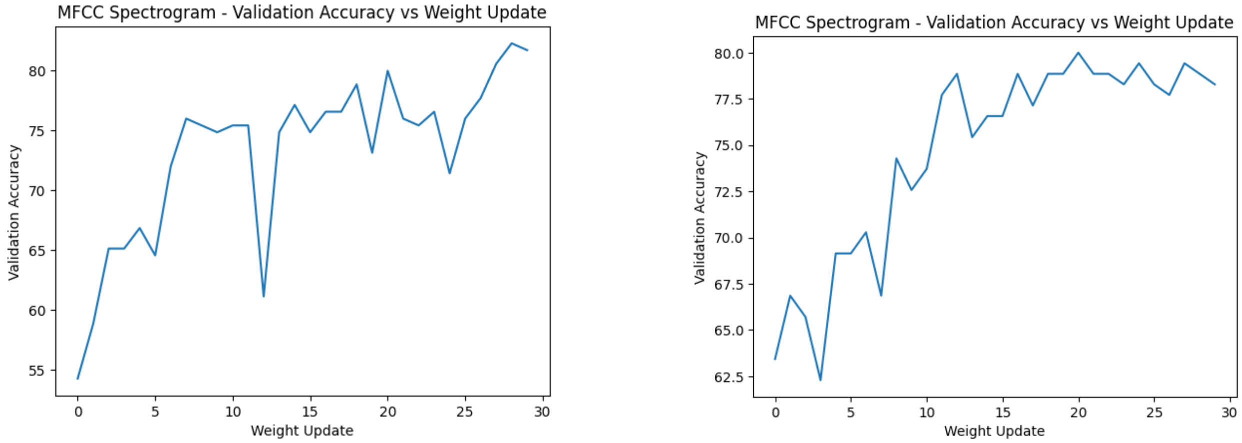
FIGURE 3. CNN Structure for MFCC data

6. EXPERIMENTAL RESULTS

6.1. **CNN Result.** We utilized both augmented and unaugmented data to trained the CNN.

6.1.1. *Pre-Augmentation CNN.* For MFCC we achieved a validation accuracy of 82% with loss that converged to near 0. The highest validation accuracy for spectrogram was 78%. The relatively high accuracies achieved for both datasets indicates that the CNN is very well-suited architecture capable to understand and classify audio via images. While the discrepancy between MFCC and spectrogram performance is small, it may suggest that MFCC representations better capture the semantic information important to classifying the sound.

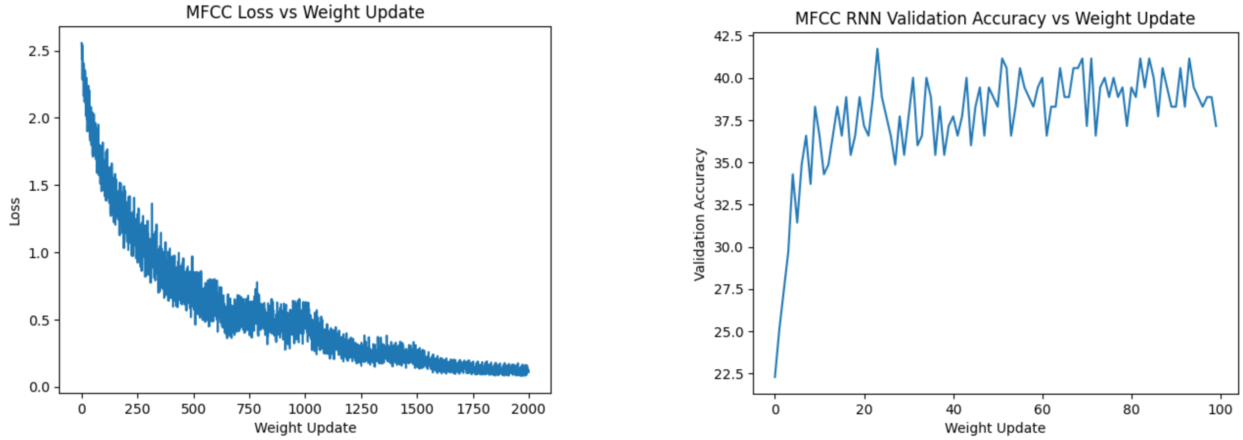
6.1.2. *Post-Augmentation CNN.* Trained on augmented data, the MFCC validation accuracy reaches 80% and spectrogram reaches 74%. The augmentation's impact is different between MFCC and spectrogram models, which means they have different sensitivity towards our augmentation techniques. For MFCC model, the accuracy drops from 82% to 80%, and for the spectrogram model, the accuracy drops from 78% to 74%. This suggests that our chosen augmentations does not help our model learn more useful features.



(A) MFCC Validation Accuracy Without Augmented Data

(B) MFCC Validation Accuracy With Augmented Data

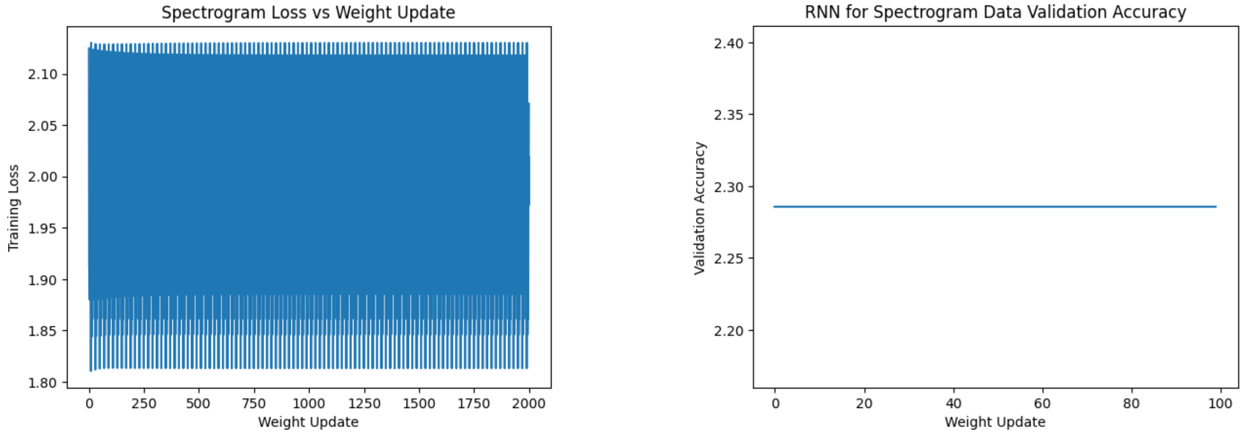
FIGURE 4. Validation accuracy on CNN trained with MFCC data



(A) RNN MFCC Training Loss

(B) MFCC validation accuracy

FIGURE 5. MFCC RNN Training Loss and Validation Accuracy without Augmented Data



(A) RNN Spectrogram Training Loss

(B) RNN Spectrogram Validation Accuracy

FIGURE 6. Spectrogram RNN Training Loss and Validation Accuracy without Augmented Data

6.2. **RNN.**

6.2.1. *RNN.* For MFCC RNN model without augmentation, the loss converges at 0.15, however the validation accuracy stagnates at 42.5%. The spectrogram model fails to converge.

6.3. **LSTM.**

6.3.1. *Pre-augmented LSTM.* The MFCC LSTM model reach 80% accuracy while spectrogram reaches only about 40%. Consistently we see higher accuracy in MFCC, demonstrating that MFCC better captures useful information. The spectrogram model has relatively low accuracy after multiple tuning of learning rate, and hidden size of LSTM. We suspect this is due to LSTM using a max-pooling trick that decrease the model's variance. We also found that LSTM with augmentation performs poorly, again suggesting that our choice of augmentation does not serve the classification task.

6.4. **AST.** For AST we only train on spectrogram as that is what AST is meant to handle. Both the pre-augmented and post-augmented AST models achieve a validation accuracy of 76%. This suggests that the model itself is a strong classifier for spectrograms, and that the augmentations do not strongly affect their performance.

7. Discussion

7.1. **MFCC vs. Spectrogram.** We consistently saw (with the exception of AST) that MFCC-based models performed better. The MFCCs have a smaller size of 13x87, which may contains more concise yet useful information about the underlying class while filtering out noise, enabling the model to predict more accurately.

7.2. **Effect of Data Augmentation.** We did not see any evidence in our experiments to suggest that our choice of data augmentations improved model performance. While this may seem to contradict Sun et al[4], data augmentation is a complicated operation. Choice of data augmentation depends largely on the dataset as certain datasets can either benefit or not benefit from certain augmentations. Automatic data augmentation itself is an active area of research, and future work could include using tools such as AutoAugment introduced by Cubuk et al[9] to search for an optimal augmentation policy to serve the dataset. Introducing data augmentation quickly increases the size of the dataset, and more thorough examination of augmentations would be suited to an environment with more computational power.

7.3. **AST vs. RNN vs. LSTM vs. CNN.** The accuracy for ASTM on MFCC is not applicable, but it works well on Spectrogram data. RNN is poor at solving both types of dataset because not capable on solving spatial information. LSTM performs better with MFCC data but dropped to hald on Spectrogram Data, meaning that LSTM face chalenge in leverage information from Spectrogram representation.
VGG achieves the highest accuracy for both MFCC and Spectrogram, as expected intilally. VGG, with a CNN architecture, is more rubust and adaptable to both types of data compared to RNN-based models. CNN-based model can learn hierarchial features, and also robust to variations. It also can identify patterns irrespective of their position. This is essential in this task because animal is not always making sound at a certain time which leads to various location of features.

	MFCC Validation Accuracy	Spectrogram Validation Accuracy
AST	N/A	72.5%
RNN	42.5%	2.3%
LSTM	80.0%	40.0%
VGG	83.0%	80.0%

References

[1] E. Şaşmaz and F. B. Tek, ”Animal Sound Classification Using A Convolutional Neural Network,” 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 625-629, doi: 10.1109/UBMK.2018.8566449.

[2] Gong, Yuan, Yu-An Chung, and James Glass. ”Ast: Audio spectrogram transformer.” arXiv preprint arXiv:2104.01778 (2021).

[3] Müller, J., Mitesser, O., Schaefer, H.M. et al. Soundscapes and deep learning enable tracking biodiversity recovery in tropical forests. Nat Commun 14, 6191 (2023). https://doi.org/10.1038/s41467-023-41693-w.

[4] Sun, Yuren, Maeda, Tatiana Midori, Solís-Lemus, Claudia, Pimentel-Alarcón, Daniel, and Buřivalová, Zuzana. Classification of animal sounds in a hyperdiverse rainforest using convolutional neural networks with data augmentation. United States: N. p., 2022. Web. doi:10.1016/j.ecolind.2022.109621.

[5] Oswald, J.N., Erbe, C., Gannon, W.L., Madhusudhana, S., Thomas, J.A. (2022). Detection and Classification Methods for Animal Sounds. In: Erbe, C., Thomas, J.A. (eds) Exploring Animal Behavior Through Sound: Volume 1. Springer, Cham. https://doi.org/10.1007/978-3-030-97540-1_8.

[6] Yosinski J, Clune J, Bengio Y, and Lipson H. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014.

[7] Zhong, Ming LeBien, Jack Campos Cerqueira, Marconi Dodhia, Rahul Lavista Ferres, Juan Velev, Julian Aide, T. Mitchell. (2020). Multispecies bioacoustics classification using transfer learning of deep convolutional neural networks with pseudo-labeling. The Journal of the Acoustical Society of America. 148. 2442-2442. 10.1121/1.5146738.

[8] Park, Daniel S. et al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition." Interspeech (2019).

[9] Cubuk, Ekin Dogus et al. "AutoAugment: Learning Augmentation Strategies From Data." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 113-123.