Type of the Paper (Article, Review, Communication, etc.)

Indian Birds Monitoring through Detection based on Audio Clippings

Tejasri Bogu 1

|  |
| --- |
| **Citation:** To be added by editorial staff during production.  Academic Editor: Firstname Lastname  Received: date  Revised: date  Accepted: date  Published: date    **Copyright:** © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). |

1 Vellore Institute of Technology, Chennai, tejasri.bn2019@vitstudent.ac.in

**Abstract:** India is rich in biodiversity but unfortunately, we are losing it due to pollution, poaching, deforestation, urbanization, and so on. Various measures are being taken to restore this loss but they aren’t enough and a lot of the steps aren’t taken based on recent data. There is a need for tracking bird species that are endemic to India, and a lot of times manually tracking them in their natural ecosystem is very hard, close to impossible especially when there are very few of them. Having an audio-based tracker can be more effective in tracking without disturbing them and is more accurate. This creates a requirement for automatic trackers and monitoring as manual monitoring is time taking, and there can be new developments even before the census is completed. Bird Identification is thus a part of this goal. There are models to identify birds but all of them work on Birds Native to America or Hawaii, having one that is for birds native to India and one that is more light, can detect both forground and background birds with and without noise is more effective. We thus compiled an Indian bird audio dataset and created a Bidirectional LSTM model that can identify birds endemic to India.

**Keywords:** Bidirectional LSTM; Bird Monitoring; Audio Classification

1. Introduction

With growing pollution, deforestation, and climate change, the degradation of the environment is at its peak. Loss of biodiversity is one such effect of the various activities by humans.

We were blessed with various bird species, each with their unique characteristics in various vibrant to dark colors playing their own part in the food cycle, but lately, their numbers have declined steeply.

Keeping track of their population, nesting areas, and other such behaviors is necessary not just for their conservation but also to identify new species. Attaining this goal without hampering their regular routine is very much important to get first-hand accurate information.

Bird watchers and conservationists have been tracking birds throughout the year, from migratory birds to ones that are sedentary. Getting precise data and numbers throughout the year, without visual contact most of the time is close to impossible and a very cumbersome task. Changing locations, lack of resources to monitor, and harsh environments of these areas make the task even harder and prone to human error.

With the development of technology of late we have surveillance systems for various use cases, setting up one for monitoring birds can solve a lot of the problems faced during manual tracking. Video and audio surveillance, motion sensors, and location trackers can be made use of to get better accurate results with the least interference in their activities.

Having continuous visuals throughout the day in all types of weather can get hard. Motion sensors can only identify in a close area, and they only alert movement. Location trackers need to be attached to the bird being tracked. Audio surveillance seems to be the most straightforward approach.

Utilizing the latest technology, we can automate identifying the bird species based on audio samples using machine learning algorithms to fast-track this entire process. There is a dire need for such a system to monitor bird species endemic and non-endemic to India.

In this project, we have collated audio samples of various birds in India to create a model to identify the species to then be used to directly track birds in real-time with the audio collected by sensors set up in their regularly frequented areas.

The existing methodologies mostly comprised of pre-trained CNN models. [1] and [2] made use of the EfficientNetB0 model with the former using Per-Channel Energy Normalization to achieve higher accuracy working only on clean data. The latter could classify fine-grained vocals but had the problem of overfitting. [3] used a CNN to classify the audio converted into an image of the spectrogram. In [4] they used AlexNet with the dataset preprocessed to remove silence along with framing and reconstruction. This model could only identify 4 species with great accuracy.

Ensemble of 13 CNN models followed by a Support Vector Machine in the next stage was used by [5]. [6] Again had the problem of overfitting using the LeNet architecture. [7] made use of and compared a variety of models from a Deep Convolutional Neural Network to pre-trained models like InceptionV3, resNet152, NasNet, and so on. They ran models individually and in an ensemble combination. Their drawback was that certain augmentation techniques applied to the dataset had a negative impact on the features. [8] emphasized on higher frequencies and made use of AlexNet but used audio from ideal environment without noise.

This paper [9] had an Ensemble of CNN and a Multi-Layer Perceptron, but either could work on foreground species only or background species. [10] used a CNN to identify foreground species.

2. Materials and Methods

* + - 1. Dataset

The dataset was created by collating audio of Indian birds from the Indian Bird Song website (https://indianbirdsong.org/). The database is sourced from Xeno-Canto avian acoustic database. There are around 1,321 avian species in India, we have recordings of 967 of them available. Out of these, we selected species with a larger base of recordings (30-35 and above per specie) to be able to train our model well with a good variety of recordings. Alphabetically birds were accessed and audio was manually downloaded. A total of 67 species’ audio was collected. Around 4,101 audios were put together.

The audios obtained are from different places in India, different environments, and recorded by different people. The length of the audio, amount of noise, and disturbance all differ. Thus giving us a dataset very close to the data we will be getting in real time when the application is put to use.

* + - 1. Data Exploration

From the audio files, we had to create a data frame with metadata to define the classes i.e., the specie, common name, scientific name, the audio ID from the xeno-canto database, and get other information like audio length and file type. All the audio files we collated were in the mp3 format, there is an inbuilt function to read and write from wav to a numpy array but for mp3 we had to define our own function using the pydub library.

The class information like specie – common name and scientific name and the ID was got from the file name, by splitting the file name string. All this was appended to the data frame.

To remove noise, we used the noisereduce library which removes noise using spectral gating that works by calculating the noise threshold for every band of frequency from the spectrogram, this threshold is then utilized to gate noise below the threshold by computing a mask. This function allows both stationary and non-stationary noise reduction, we used non-stationary noise reduction, which allows the noise gate to be flexible and change with time.

Following this we explored various features and visualized them this included the following, time domain features - the waveform, spectrogram, RMSE, mel spectrogram, zero crossing rate, the harmonic and percussive signals, the beats, frequency domain features – chromagram, constant Q-transform, chroma energy distribution of normalized statistics, spectrum domain features- spectral centroids, spectral contrast, spectral rolloff, and cepstral domain features- mel-frequency cepstral coefficients.

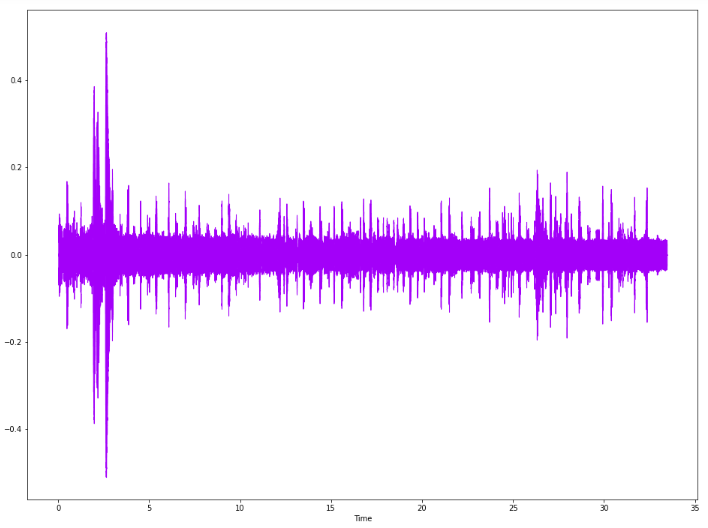


Figure Waveform of a Casvir bird audio file

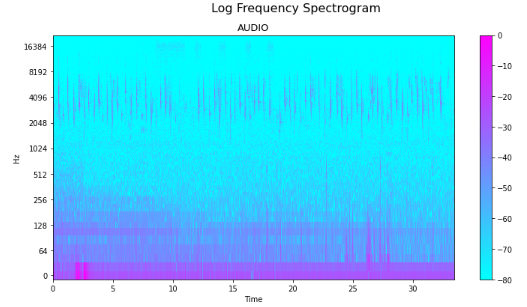


Figure Log Frequency Spectrum of a Casvir bird audio

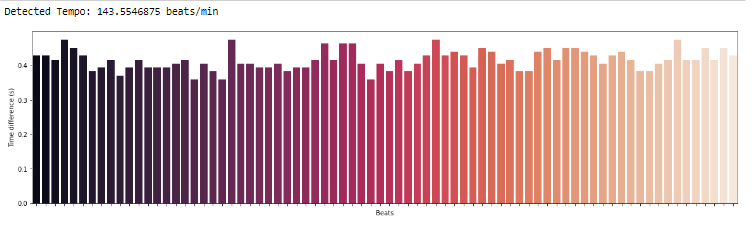


Figure Beats and Tempo of Casvir bird audio file

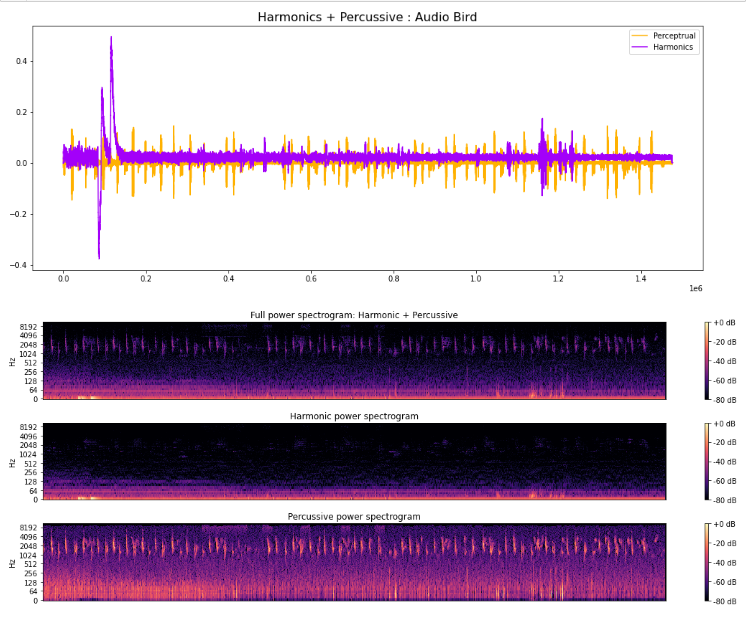


Figure Harmonics and Percussive wave and spectrogram of Casvir bird

Figures 1-4 show various features explored. For easier understanding an audio of the Casvir bird has been shown here.

* + - 1. PreProcessing - 1

The audio files put together were denoised. Both the noisy and noiseless audio are used to train, test and validate the model. Each file is considered a separate instance, so initially, a single file is denoised, and then both the noiseless and the noisy file are used as 2 separate instances of the same class. So a total of 8,202 audios are used.

* + - 1. Feature Extraction

Following denoising, feature extraction was performed. Multiple features were extracted over various domains. Spectral Centroids, RMS, Chromagram, and MFCCS features are extracted.

The spectral domain refers to the representation of an audio signal in terms of its frequency content. In the spectral domain, the signal is transformed from the time domain into a frequency representation, such as a spectrogram or power spectrum. Spectral Features describe the vibration amount at every frequency. Each individual bird has a unique combination of frequencies for their tune, recognizing this sequence can be used to identify a bird thus making it a valid feature for training.

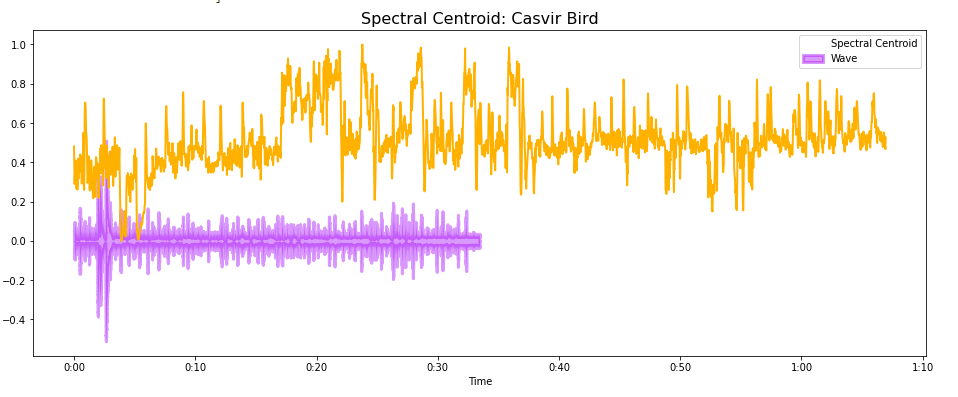


Figure Spectral Centroid wave form of Casvir bird audio

In the spectral domain, the first feature was extracted, the Spectral Centroid. It aims to find the center of mass of the audio, it is connected to the timbre or tone quality. For each frame, the mean is calculated after normalizing the magnitude spectrum. The frequency bins generally considered use Fourier Frequencies. Using the librosa library we were able to find the spectral centroids with a sampling rate of 44,100 and Short-time Fourier Transform (STFT) hop length of 8,192 for the returned feature array length to be kept less than 1,000.

The time domain refers to representing a signal as a function of time, typically as a waveform. This representation of a sound provides information about the evolution of the amplitude of the sound over time and is particularly useful for analyzing the temporal characteristics of a sound, such as its attack, sustain, and release.

In the time domain, the RMS (Root Mean Square) is the feature that was calculated. It computes the average loudness for the audio in other words for a waveform it is the net effective value. For each specie, this value differs thus making it a good parameter to base our classification. For each frame the RMS is found from the spectrogram, as even though this takes a bit more time for computation of the STFT it is more accurate in representing the energy in time over frames as it can use windows. Again, a hop length of 8,192 and a sampling rate of 44,100 was used.

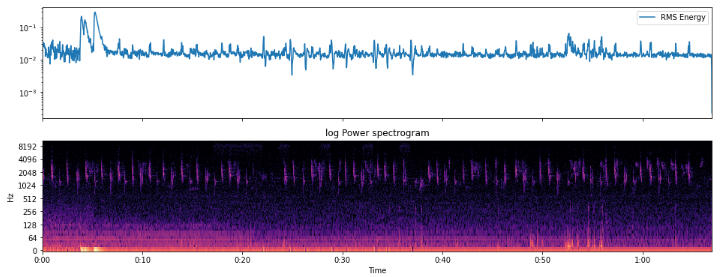


Figure RMSE and Log Power spectrum of Casvir bird audio

The chroma domain refers to a representation of an audio signal in terms of its pitch content, rather than its time or frequency content. In the chroma domain, the signal is transformed into a set of chroma features, which capture the presence of different pitch classes in music.

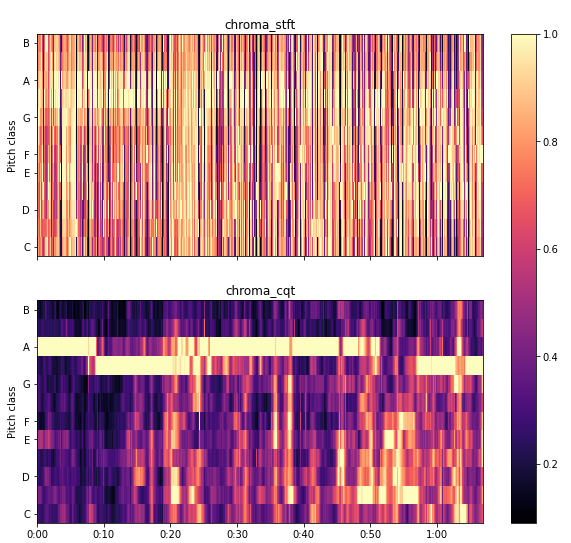


Figure Chromagram and Q-transform Chromagram of Casvir bird audio file

The Chroma feature is another feature we used. It converts the values of the magnitude spectrum into 12 chroma or semitones i.e., 12 bins. It is basically the pitch-class profile. The melodic and harmonic characters of the audio clip are identified by this feature. Species have their own melody to their bird songs, and identifying what is unique to them can contribute to identifying the specie. This feature is robust to change in the timbre and harmony and can thus identify the difference in octaves of pitches. STFT is again made use of to calculate the chromograms, we used a hop length of 8,192 and a sampling rate of 44,100.

The cepstral domain refers to a representation of an audio signal in which the log-spectrum has been taken and then transformed back into the time domain. In this domain, the audio signal is transformed from the frequency domain into a representation that is based on the human perception of sound.

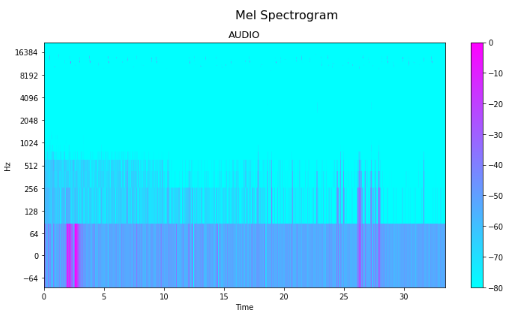


Figure Mel Spectogram of Casvir bird audio file

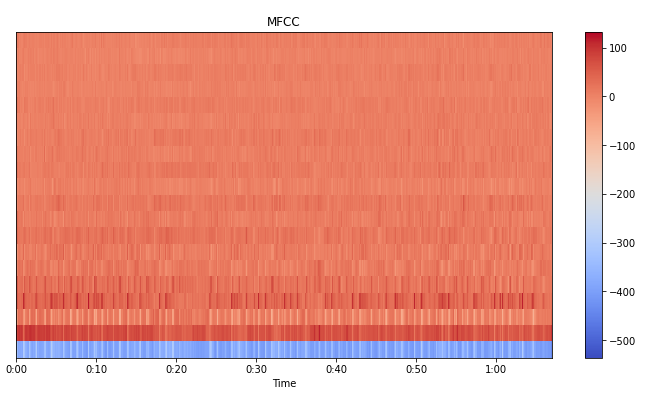


Figure MFCCS of Casvir bird audio file

The final feature taken was the Mel-Frequency Cepstral Coefficients (MFCCs). We used MFCCs as it shows the information in the signal that mimics human hearing perception, low-level auditory, letting us thus add that small human hearing aspect. Technically, the MFCC is a representation on the Mel scale, as human perception of frequencies is not linear. On this scale, the pitch that sounds to have equal distance to humans is equally distant. An STFT spectrogram is again used here, a hop length of 8,192 and a sampling rate of 44,100 was used by us. The spectrogram is then represented in the Mel scale by applying mel scale filter banks, this spectrogram is called Mel spectrogram. A Discrete Cosine Transform (DCT) is then applied to this to get the MFCCs. Normally the first 2 to 13 coefficients are taken of the signal after the transform, and 26 features that include coefficients of derivatives of first and second order giving a total of 39 features, but for this project, we used 125 features, the finer details of the spectrum are represented by the higher coefficients.

* + - 1. PreProcessing - 2

The extracted features are then padded with zeros or trimmed depending on their size. A maximum length of 1000 is applied on all features, to equalize the dataset. Around 64 instances had their features trimmed, rest had them padded with zeroes.

Since different features have different ranges we also created a normalized dataset using nomalize\_2D that works by converting into a unit vector.

Following this the labels are encoded using LabelEncoder, that is here the labels are normalized. The non-numerical classes here ( the specie names) are converted into numerical classes to transform the data to be able to pass it to the model. Here now each class is assigned a number from 0 to 67.

The dataset is now split into train, test and validation with 64% , 16%, 20% of the data giving 5,250 instances to train, 1,641 to validate and 1,313 to test the model on.

* + - 1. Model

We initially tried a variety of models on a smaller dataset with just 38 classes to find the one best suited for this task. A simple Neural Network , simple Convolutional Neural Network (CNN) , pre-trained CNN models like Inception, ResNet50, VGG16, and EfficientNet were used with Stochastic Gradient Descent, Adagrad, and RMS as optimizers. The accuracy was very marginal, the highest being 35.8% for the Neural Network with the Stochastic Gradient Descent optimizer.

An Artificial Neural Network (ANN) works well with textual or tabular data, when compared to a CNN or an RNN it is less powerful, and also has the problem of vanishing gradient. CNN works well with image data here using a sequence is easier, and more features are extracted, also it is spatially invariant i.e., when we shift/move the signal at input there is an equal output signal shift. When following this, RNN seemed to be the better solution. To work on the long-term dependence and the vanishing gradient problem, we used Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM), instead to improve the memory.

An LSTM has both feedforward and unlike other standard feedforward neural networks’ feedback connections too. This allows us to use a chain of data instead of single points. LSTM passes important information through the network and ignores unnecessary information. It makes use of its three different gates to do so, the input, the forget, and the output gates. Although an LSTM does solve the vanishing gradient problem to some existent, it still exists. The sequence length that it can remember is still short and the path traveled is a long one.

A GRU makes use of reset and update gates to store information. They can keep very old information without diluting it over time while adding only necessary information and removing unnecessary information.

A BiLSTM has two LSTMs, one to take the input forward and one backward. The quantity of information available to the network is increased, thus the context given to the algorithm is improved drastically. Sequential dependencies are thus modelled well. Using mean, sum, concatenation or multiplication the outputs are combined from both the layers. When compared to GRUs and LSTMs, the BiLSTM can store more data and can use both past and future information in a particular time step.

When trained on our dataset a GRU based model did not give us satisfying results, so we tried a combination of LSTM and Bidirectional LSTM layers for our model.

3. Results

For convenience, we will only be discussing the models with decent results. Figures 1-4 shows the variety of architectures we tried that gave significantly good results. We used a masking layer to have the variable input size of the audio clips corrected.

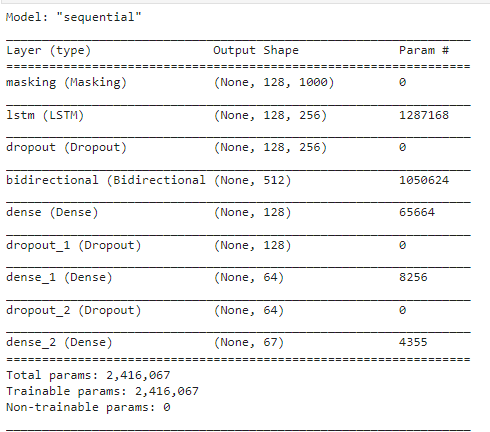


Figure Model 1

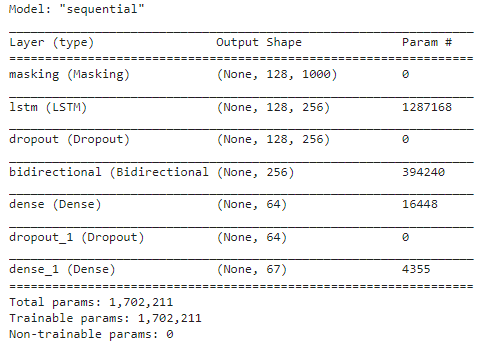


Figure Model 4

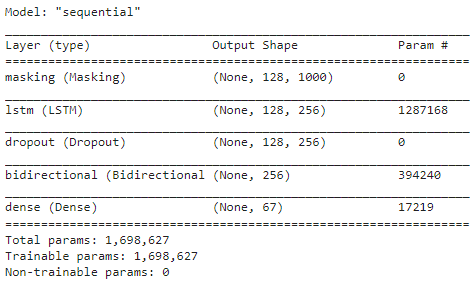


Figure Model 5

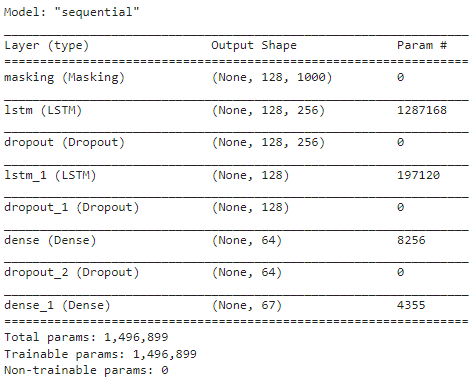


Figure Model 7

1. Model 1

Figure 1 (Model 1) has an LSTM layer, followed by a dropout layer to prevent overfitting. We used a Bidirectional LSTM layer with three dense layers.

1. Model 4

Figure 2(Model 4) is a very simple model with a masking layer, LSTM, Dropout, Bidirectional, Dense, Dropout and Dense.

1. Model 5

The next model in Figure 3 (Model 5) is similar to Model 4 with only one final dense layer after the Bidirectional layer.

1. Model 7

This model in Figure 4 (Model 7) is a pure LSTM model without a Bidirectional LSTM Layer. With LSTM and 3 Dropout and Dense layers.

**Table 1** Model Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Epochs | Trainable Parameters | Accuracy | Loss |
| Model 1 | 50 | 2,416,067 | 0.9730 | 0.0885 |
| Model 4 | 50 | 1,702,211 | 0.9840 | 0.0581 |
| Model 5 | 50 | 1,698,627 | 0.9998 | 0.0023 |
| Model 7 | 50 | 1,496,899 | 0.9790 | 0.0717 |
| Model 4 with Normalized Data | 200 | 2,416,067 | 0.9600 | 0.1300 |

Table 1 shows the results obtained by fitting the model on train dataset and testing and validating the trained model. Model 1 has the highest parameters that had to be trained, thus having a very high train time but ended up having the least accuracy. Model 7 has the least trainable parameters and a pretty low accuracy too. Both of them have a significantly higher loss too when compared with the other two models. Model 4 is in between with the Accuracy for train and validation along with a pretty low loss.

We then took Model 4 and trained it using the normalized data. Results seemed to improve. The accuracy fell a little but the validation of the model improved significantly.

4. Conclusions

We thus observed that using a combination of Biidirectional LSTM model with LSTM with few Dense and Dropout layers over normalized audio feature data using the features Spectral Centroids, RMS, Chromagram, and MFCCS extracted from both noisy and noiseless audio files seemed to work the best. For future work we can work on expanding the dataset with more records per bird and expanding to more rarer birds in India.

References

[1] Mark Anderson & Naomi Harte (2022). Learnable Acoustic Frontends In Bird Activity Detection

[2] Marcos V. Conde & Ui-Jin Choi (2022). Few-shot Long-Tailed Bird Audio Recognition

[3] Pooja Wale, Abhishek Mankar, Pratik Padale, Sanket Gawade & Prasanna Ghogare (2022). A Survey on Bird Species Identification Using Audio Signal Processing and Neural Network

[4] Amol Dhakne, Vaishnav M. Kuduvan, Aniket Palhade, Tarun Kanjwan & Rushikesh Kshirsagar (2022). Bird Species Identification using Audio Signal Processing and Neural Networks

[5] Marcos V. Conde, Kumar Shubham, Prateek Agnihotri, Nitin D. Movva, Szilard Bessenyei Marcos V. Conde, Kumar Shubha, Prateek Agnihotr, Nitin D. Movva & Szilard Bessenyei (2021). Weakly-Supervised Classification and Detection of Bird Sounds in the Wild. A BirdCLEF 2021 Solution

[6] Shirley Cheng & Julie Wang (2021). Detection of Bird Species Through Sounds

[7] Mario Lasseck (2020). Audio-based Bird Species Identification with Deep Convolutional Neural Networks

[8] Chandu B, Akash Munikoti, Karthik S Murthy, Ganesh Murthy V & Chaitra Nagaraj (2020). Automated Bird Species Identification using Audio Signal Processing and Neural Networks

[9] Jan Schlüter (2019). Bird Identification from Timestamped, Geotagged Audio Recordings

[10] Elias Sprengel, Martin Jaggi, Yannic Kilcher & Thomas Hofmann (2018). Audio Based Bird Species Identification using Deep Learning Techniques

[11] XIE Jiang-jian, DING Chang-qing, LI Wen-bin & CAI Cheng-hao (2018). Audio-only Bird Species Automated Identification Method with Limited Training Data Based on Multi-Channel Deep Convolutional Neural Networks

[12] Rafael H.D. Zottesso, Yandre M.G. Costaa, Diego Bertolini & Luiz E.S. Oliveira (2018). Bird species identification using spectrogram and dissimilarity approach

[13] Stefan Kahl, Thomas Wilhelm-Stein, Holger Klinck, Danny Kowerk & Maximilian Eibl (2018). Recognizing Birds from Sound - The 2018 BirdCLEF Baseline System

[14] Emre C¸ akır, Sharath Adavanne, Giambattista Parascandolo, Konstantinos Drossos & Tuomas Virtanen (2017). Convolutional Recurrent Neural Networks for Bird Audio Detection

[15] Sharath Adavanne, Konstantinos Drossos, Emre C¸akır& Tuomas Virtanen (2017). Stacked Convolutional and Recurrent Neural Networks for Bird Audio Detection