EFFECTS OF FREQUENCY BAND REMOVAL ON MODEL ACCURACY

Project Echo

Abstract

In this report, I will digest deeper into investigating the optimized frequency range to train audio that can maintain high accuracy. The experiments will be conducted by testing the model with different range of frequency

Nhat Minh Dang

222172836

Contents

A) Introduction:

Project Echo was founded in Trimester 3 of 2022 by Stephan Kokkas, Andrew Kudilczak and Daniel Gladman. The project aims to support global conservationists by developing advanced audio classification systems that can facilitate the non-intrusive monitoring, discovery and tracking endangered species and their predators within their natural habitats. The system of IoT will collect data and transmit to a center server via an API, where an AI-driven model classifies species and record vital data. Then, the Human Machine Interface (provides real time visualization of simulated animals and vocalization events on an interactive map.

As the project develops, more and more audio data are collected, increasing the current capacity. These audio files are varied in frequency, quality, duration, etc. In this task, I will digest deeper into optimizing the frequency and save some space for new audio datasets. First, the audio samples will be trained through different models to see which is the model with best accuracy. Then, the datasets will be optimized through different Frequency Band Removal Methods with different range of frequency, followed by training using those best models to see which one shows the best result.

B) Definitions and Concepts:

1) Frequency in Audio:

Frequency refers to the number of sound wave cycles that occur per second, measured in Hertz (Hz). It originates from the physical vibrations of an object, such as vocal cords, instruments, or environmental sources, which cause air molecules to oscillate and produce sound waves. These vibrations vary in speed; slower vibrations result in lower frequencies (e.g., deep growls), while faster vibrations create higher frequencies (e.g., bird chirps).

Frequency is a critical feature in audio analysis because it captures the tonal and spectral characteristics unique to different sound sources, including animal vocalizations. In animal audio classification, frequency patterns play a vital role in distinguishing species or behaviours, as many animals produce sounds within specific frequency ranges that reflect their physical structure or communication needs. Therefore, optimizing classification models with specific frequency bands helps isolate meaningful signal components and reduces noise or irrelevant data. For instance, removing irrelevant high frequencies in low-pitched animal calls can improve model focus and accuracy, while retaining key bands ensures that the classifier captures the essential acoustic signatures unique to each animal. This optimization is essential in creating robust and efficient audio classification systems.

Below is the picture of Audio Spectrum (**Figure 1**), which is divided into specific ranges. Each field is designed with different purpose:

- Sub bass: Spanning from 20 Hz to 60 Hz. The sound is deep and resonant, which is often felt rather than heard. Kick drums and bass synthesizers are examples for this.
- Lower Midrange: Covering 250 Hz to 500 Hz, the bass segment provides warmth and substance to sound. This range is less perceptible to the human ear, requiring strong amplification for precise reproduction. This includes instruments like bass guitars and lower piano registers, which anchor the overall sound structure.

- Midrange: From 500 Hz to 2 kHz, the midrange is popular for vocal articulation and the distinct separation of instruments within a max.
- Upper midrange: Spanning from 2 kHz to 4 kHz, the upper midrange contains the harmonics of lower pitched instruments, significantly affecting clarity and timbre and requires heightened sensitivity in human hearing.
- Presence: Ranging between 4 to 6 kHz, it is a demonstration of high pitch sounds from instruments like violins and guitars.
- Brilliance: From 6 kHz to 20 kHz, the brilliance range captures high frequency details that bring sparkle and life to audio, followed by adding ethereal quality to music.

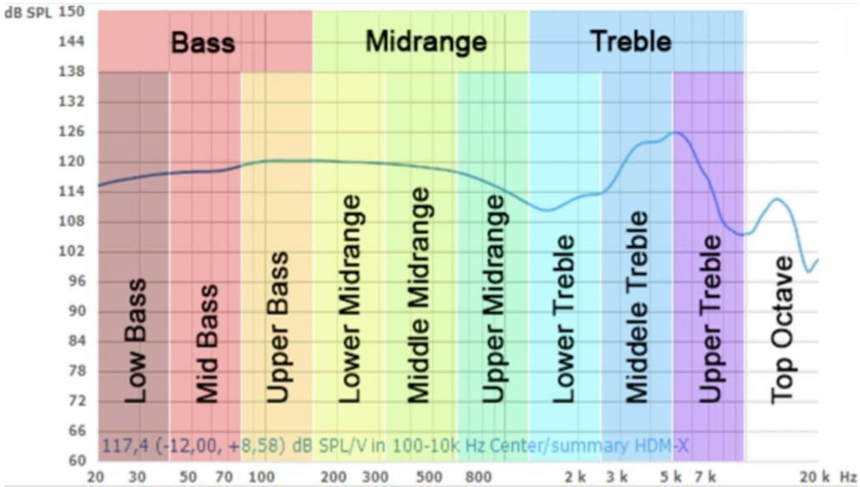


Figure 1: Audio Spectrum.

2) Frequency Band Removal:

Frequency Band Removal is the process of selectively attenuating or removing specific ranges of frequencies from and audio signals or its spectral representation. This is also know for another term called **Band Rejection.** I will use this method to study the importance of different frequency ranges for animal audio classification task, which improves robustness again noise and simulate real world scenarios where certain key components may be lost or distorted. In audio processing, frequency band removal will help identify the most informative features of a signal and optimize the models by focusing on critical frequency range and reduce irrelevant or redundant data. These are the band removal methods that I will test.

a) Bandstop Filtering:

Bandstop Filtering is a signal processing technique that removes or attenuates specific frequency range from audio signal while preserving the frequencies outside the range. In mathematical description, bandstop filter can be demonstrated as a combination of low pass and high pass filter if the bandwidth is wide enough that two filters do not interact too much. This is particularly used for removing unwanted noise or interference such as hums or tones from electrical equipment that could distort the audio features critical for classification task.

In this audio classification tasks, bandstop filtering is used during the preprocessing stage to ensure that the input signal retains the relevant frequency components for accurate feature extraction (Mel frequency cepstral coefficients (MFCCs) or spectrogram analysis. The filtered signal can help models to focus on meaningful patterns without being misled or noisy data, improving accuracy and robustness. There are several parameters in bandstop filter, but I will only use the frequency range for simpleness. The pictures below represent the generic idea of bandstop filter, where a notch filter is a band-stop filter with a narrow stopband. The transfer function is below with ω_0 is the central rejected frequency and ω_c is the width of the rejected band:

$$H(s) = \frac{s^2 + \omega_0^2}{s^2 + \omega_c s + \omega_0^2},$$

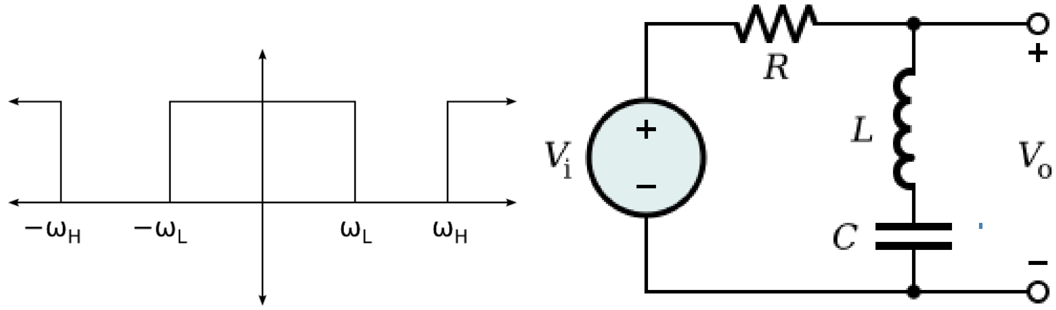


Figure 2: Visualization of Idea for Band Stop Filter.

In this experiment, I will use the Butterworth Filter, one of Bandstop filter Method that works by flattening frequency to 0. The Butterworth filter is a type of signal processing filter designed to have a maximally flat frequency response in the passband, ensuring no ripples and minimal distortion. It is widely used in audio, communication, and control systems where smooth frequency attenuation is crucial. Named after physicist Stephen Butterworth, it was introduced in 1930 as an ideal solution for applications requiring a flat response. The Butterworth filter is characterized by its order, which determines the sharpness of the transition between passband and stopband; higher orders result in steeper roll-offs. Implemented in various forms, such as low-pass, high-pass, bandpass, or bandstop, it uses mathematical functions to shape the frequency response. Its straightforward design and effectiveness make it a popular choice for applications needing high-quality signal filtering without introducing excessive phase distortion.

b) Bandpass Filtering:

Unlike Bandstop filter, Bandpass filter is a signal processing tool that is designed to isolate a specific range of frequencies, allowing only those frequencies to pass. Meanwhile, the outside range is attenuated. This is valuable in scenarios where only certain frequency components are relevant for classification such as distinguish between speech and background noise or identify specific musical instruments.

In audio classification, bandpass filtering is often applied during preprocessing to focus on frequency bands that contain critical information such as formants in speech or harmonic structures in music. By narrowing down the analysis to relevant frequencies, a bandpass filter helps reduce noise and irrelevant features, leading to cleaner input for feature extraction techniques like Mel-frequency cepstral coefficients (MFCCs) or spectrogram generation. The filter's parameters, which includes the lower and upper cutoff frequencies and the filter order, are typically chosen based on the characteristics of the audio signal and the classification task. Outside of engineering, bandpass filter can be used to extract the business cycle component in economic time series, 4G and 5G wireless communication with noise removed, energy scavengers, etc. Below is the concept of Bandpass filter:

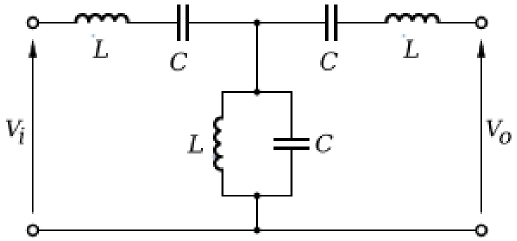


Figure 3: Idea of Bandpass filter.

3) Yam-Net Model:

YAMNet (also known as Yet Another Mobile Network) is a pre trained deep learning model that is designed for audio event detection and classification. YAMNet is developed by Google and based on the MobileNetV1 architecture and uses audio features like log Mel spectrograms to recognize 521 sound classes from the Audio Set dataset such as speech, animal, environment noises. Because of its lightweight design, I have chosen this model for its efficient in mobile devices, followed by it sparse range of classes.

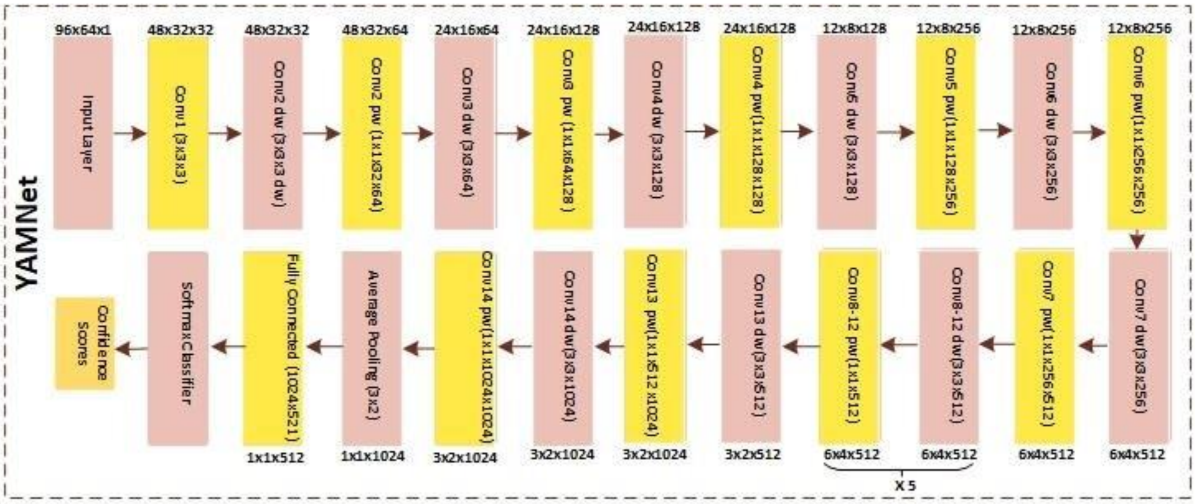


Figure 4: YAMNet architecture.

Structure of YAMNet model:

- Input Layer: Accepts log-Mel spectrogram with a shape of (96, 64) where 96 is time frame and 64 is the Mel frequency bands.
- Feature Extractor: A series of depth wise separable convolutional layers from the MobileNetV1 architecture. These layers are efficient and suitable for real-time applications, capturing the spatial and temporal patterns in the spectrogram.
- Global Average Pooling: Aggregates feature maps into a single vector, reducing the dimensionality while retaining the most important information.
- Full Connected Layer: Maps the extracted features to the output classes, producing the probability distribution over the 521 classes.

- Softmax Output: Provides class probabilities, enabling multi-label classification for overlapping sound events.

How does YAMNet work:

- Input representation: YAMNet processes audio as log-Mel spectrograms, which are compact representations of sound energy across time and frequency. The input to the model is typically a 1-second audio clip, sampled at 16 kHz, converted to a 64-band Mel spectrogram.
- Feature Extraction: The Mel spectrogram is fed into the model, where convolutional layers extract hierarchical features. These features represent the time-frequency patterns associated with different sound classes.
- Classification: YAMNet outputs a probability distribution over 521 sound event classes from the Audio Set ontology. Each probability represents the likelihood that a specific sound event is present in the input audio.
- Post Processing: Users can threshold or aggregate these probabilities over time to detect and classify sound events in longer audio clips.

C) Set up and Configuration:

1) Datasets and evaluation:

Based on my knowledge, each animal has its own sound features. While some of them have high frequency (bird, mouse, cat), there are others that have low frequency (cow, lion, tiger). Therefore, I will test each animal separately on the YAMNet model. For the evaluation, I will focus mostly on the model accuracy, which is measured by number of either correct prediction (on specific animal species) or animal recognitions only. Below is the list of animals and its expected prediction (other than Animal prediction):

- Aslan: Roaring cats (lions, tigers): Lion.
- Cat: Roaring cats (lions, tigers).
- Dog: Canidae, dogs, wolves.
- Koyun: Sheep.
- Kurbaga: Frog.

2) Processing steps:

In general, the data will be trained with YAMNet as normal to see the standard accuracy that the model will acquire for that animal. Then, I will perform Bandstop Filter to find feature patterns for an animal. Finally, to confirm my finding, I will use Bandpass Filter to perform prediction using the only found features.

All the data files are in .wav format. To preprocess the data, after loading data, I set the sample rate to 16 kHz. Then, I normalize the values between -1 and 1. The processed waveform will be sent as input into the model. Once getting the scores and predicted classes, I will retrieve the top class and top probabilities to add it into the result. When all data gets its result, the accuracy is calculated by the number of correctly predicted classes divided by the total number of prediction (equal to the total number of files loaded).

The data will be trained again with Bandstop filter. By removing each 100 Hz part of the audio files, I can inspect the frequency parts that affects the accuracy. Using the same preprocessing steps and model, the accuracy will be collected to draw a graph. The visualization demonstrates the impact of removed frequency to the model accuracy. Once the feature range is confirmed, I will use Bandpass filter, leaving the expected range or the other range to test the overall accuracy. These are the important attributes that needs to be modified for smoothest performance:

- Audio directory path to your audio files (must be in .wav format).
- Desired class (refer to the class map of the model).
- Minimum and maximum range of Bandstop filtering to cut (min_range and max_range).
- Distance between each cut.
- Minimum and maximum range of Bandpass filter to cut (min_test and max_test).
- Distance between each test cut.

3) Visualization:

These are the visualization that I will be used for displaying audio files and results

a) Waveform:

The waveform visualization depicts the amplitude of the audio signal over time, providing a snapshot of its loudness dynamics. Peaks in the waveform indicate high-amplitude sections, which correspond to louder sounds, while troughs show quieter moments. This visualization is especially useful for identifying temporal features such as when certain events occur, sudden changes in volume, or periods of silence. For example, in speech, there might be some distinct pauses between words or sentences, whereas in music, rhythmic beats or sudden crescendos might stand out. Waveforms provide an intuitive way to assess the overall structure and loudness variation of the audio file.

b) Log Mel Spectrogram:

The log Mel spectrogram transforms the audio signal into a time-frequency representation, showing how the intensity of frequencies evolves over time on a perceptual Mel scale. This visualization highlights both the pitch and timbre of the audio, with horizontal bands indicating dominant frequencies or tones. Darker regions signify areas of lower intensity, while brighter ones reflect high-energy frequencies. The log scaling emphasizes lower frequencies, making it easier to analyze speech patterns, musical notes, or bass-heavy components. By visualizing the time and frequency domains together, the spectrogram is particularly valuable for understanding complex audio features, such as phonemes in speech or chord progressions in music.

c) Frequency Spectrogram:

The frequency spectrum offers a static view of the distribution of frequencies within the audio, showing the amplitude of each frequency component. By analysing the peaks, user can determine which frequencies dominate the signal, such as low frequencies for bass notes or high frequencies for treble and harmonics. This visualization is ideal for evaluating the tonal balance of the audio, whether it is bass-heavy, mid-range-focused, or treble-rich. It also reveals harmonic relationships, which are common in both speech and musical recordings. The frequency spectrum complements the other two visualizations by providing a clear, overall summary of the spectral content.

d) Line Graph:

To show the relationship between missing frequency part with model accuracy, I use this graph to demonstrate it.

D) Result:

1) Aslan (Lion):

- Desired class: Roaring cats (lion, tigers); Animals.

The lion is a majestic big cat commonly referred to as the "king of the jungle," though it typically inhabits savannas, grasslands, and open woodlands rather than dense forests. Native to Africa and a small region in India, lions are social animals that live in groups called prides. In audio classification, the lion's roar is a key feature often analysed, characterized by its low-frequency rumble, high amplitude, and significant volume, capable of traveling up to 8 kilometres. These acoustic features make lion roars distinct and are crucial in identifying the species in bioacoustics studies, enabling researchers to monitor populations and behaviours in their natural habitat. For this animal, I have 45 audio samples for classification. Below is an example and visualization of audio file when predicting with the model.

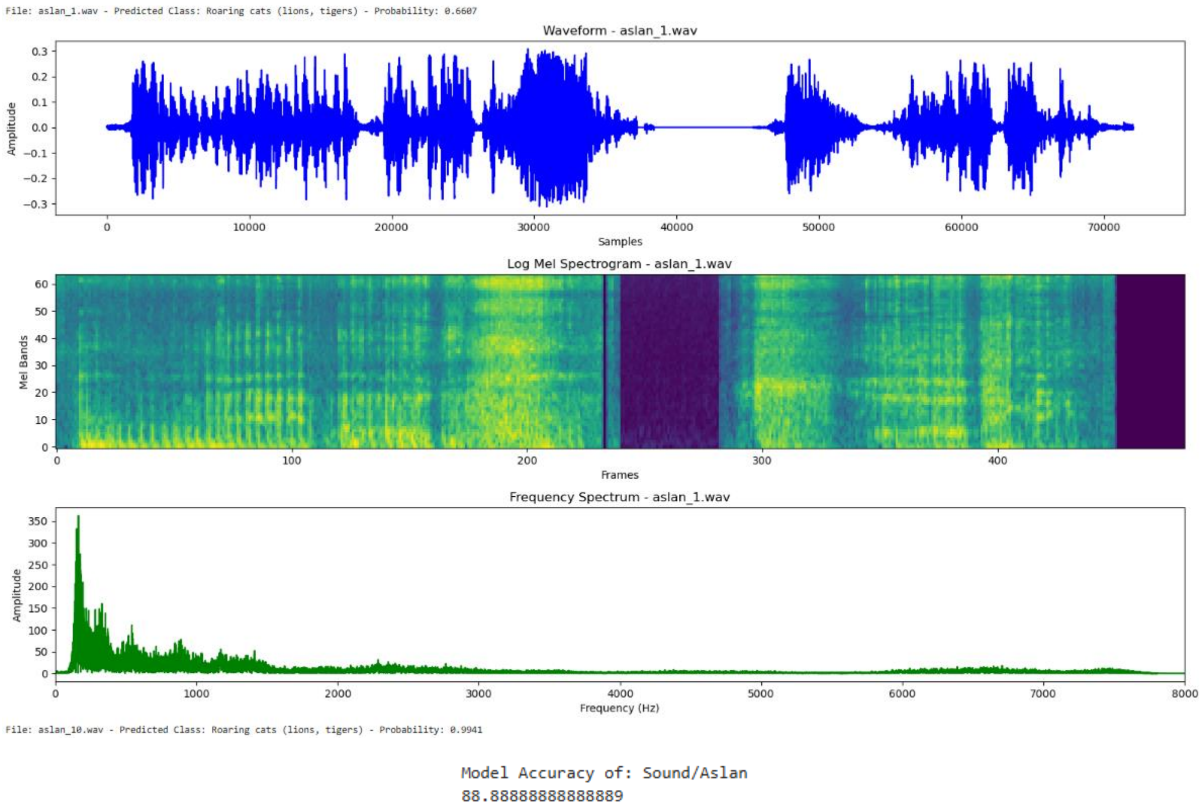


Figure 5: Example and base result of Aslan.

In this example, the roar of lion is divided in two stages. In each part, the amplitude reaches maximum at nearly 0.3 in the first one and 0.25 in the second one. While the duration is shorter in the second stage, the first stage shows more intense in Mel bands. However, in both stages, the frequency of lion mostly focuses on below 1000. To inspect further into the result, I will use Bandstop to inspect the changes of accuracy with missing part of the audio files.

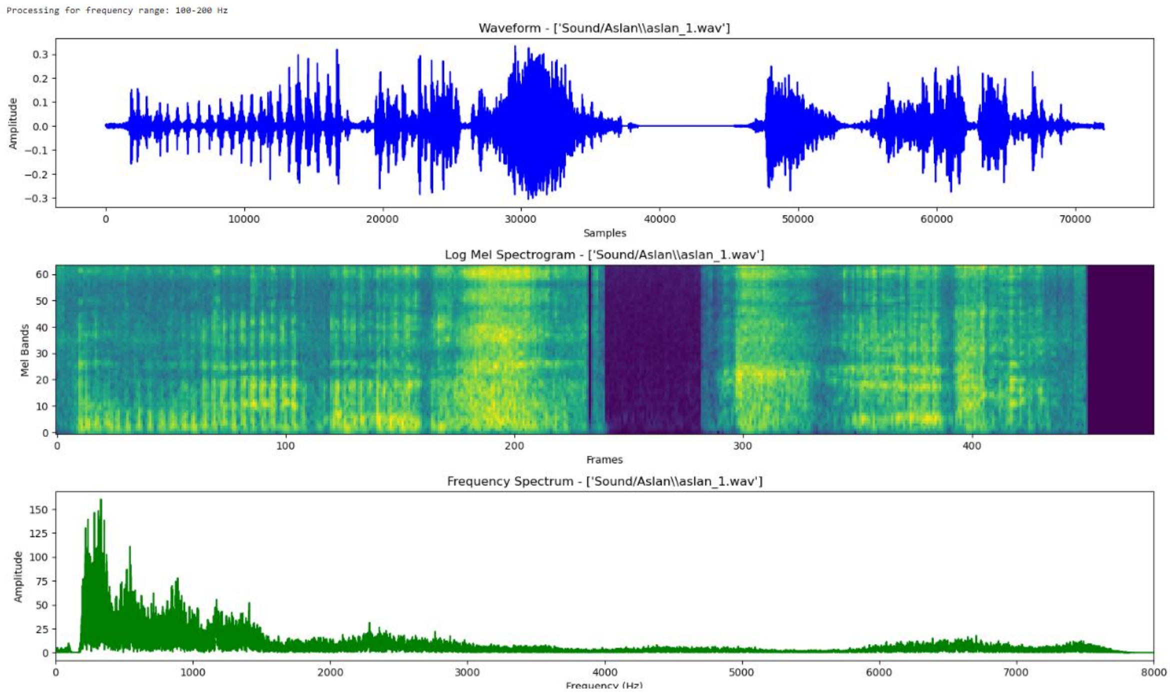
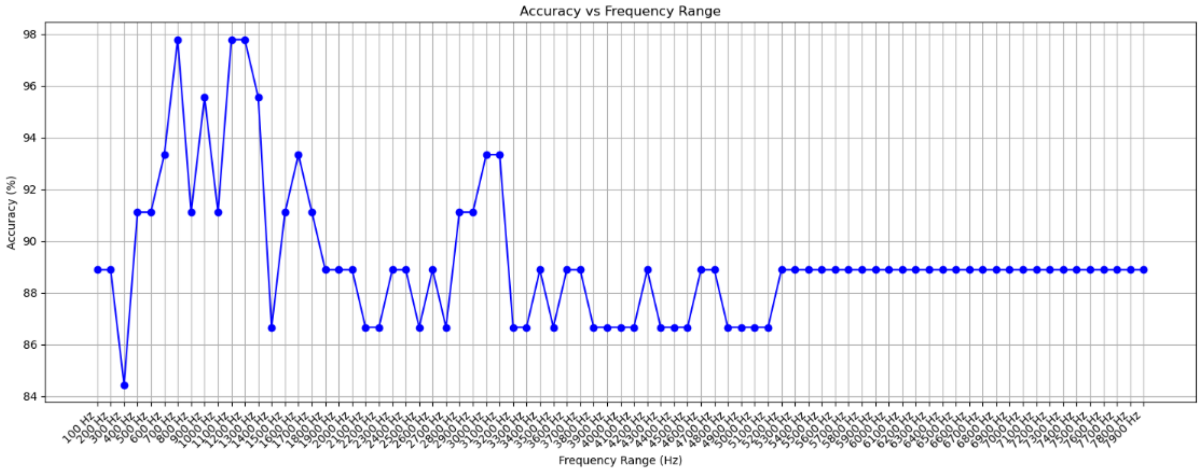


Figure 6: A significant example of using Bandstop filter.

With the use of Bandstop filter between 100 Hz to 200 Hz, there are some notably differences can be acquired. In the waveform graph, the loudness decreases significantly by 0.2 in the first part of the audio. Furthermore, as can be seen from the Log Mel Spectrogram and Frequency Spectrum, the amplitude also declines dramatically from 350 to 150. Therefore, the weight of frequency between 6000 to 8000 increases when training the model, followed by learning this irrelevant frequency more as part of feature.

```
   Frequency Range  Accuracy (%)
0           100 Hz     88.888889
1           200 Hz     88.888889
2           300 Hz     84.444444
3           400 Hz     91.111111
4           500 Hz     91.111111
..             ...           ...
74         7500 Hz     88.888889
75         7600 Hz     88.888889
76         7700 Hz     88.888889
77         7800 Hz     88.888889
78         7900 Hz     88.888889

[79 rows x 2 columns]
```




```
Processing for frequency range: 0-3000 Hz
Processing for frequency range: 3000-6000 Hz
  Frequency Range  Accuracy (%)
0         3000 Hz     88.888889
1         6000 Hz     22.222222
```

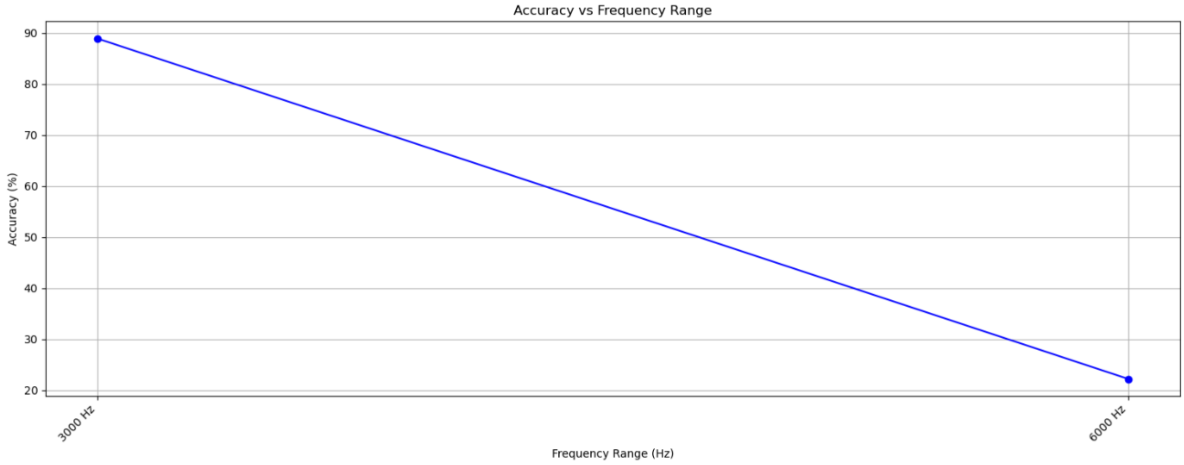


Figure 7: Changes of accuracy based on missing frequency part for Aslan using Bandstop (top) and Bandpass (bottom) filter.

As can be seen from the result, the accuracy changes mostly only between 200 Hz to 3200 Hz, followed by small fluctuation from 3200 Hz to 5300 Hz. Significantly, removed frequency parts 700 Hz to 800 Hz and 1100 Hz to 1300 Hz increase the accuracy to nearly 98%. Therefore, it can be concluded that those are the noise part of the sound when extracting feature. To confirm again, with the result of Bandpass filter. It can be demonstrated that the feature frequency for lion is between 200 Hz to 3000 Hz, with ranges 700 Hz to 800 Hz and 1100 to 1300 Hz are the noise frequency range.

2) Cat:

- Desired class: Cat; Roaring cats (lion, tigers) (optional); Domestic Animals, pets; Animals

Unlike Aslan (lion), the audio files of cats are quite interesting. To be specific, this is mostly considered a domestic animal, raising by human. Therefore, the sound of cats is affected by their living environment, changing the frequency significantly but mostly have higher frequency than lions. As a result, the prediction is more likely to be classified as human activities like speech, crying, sobbing, etc. With 200 audio samples, this is what I have found:

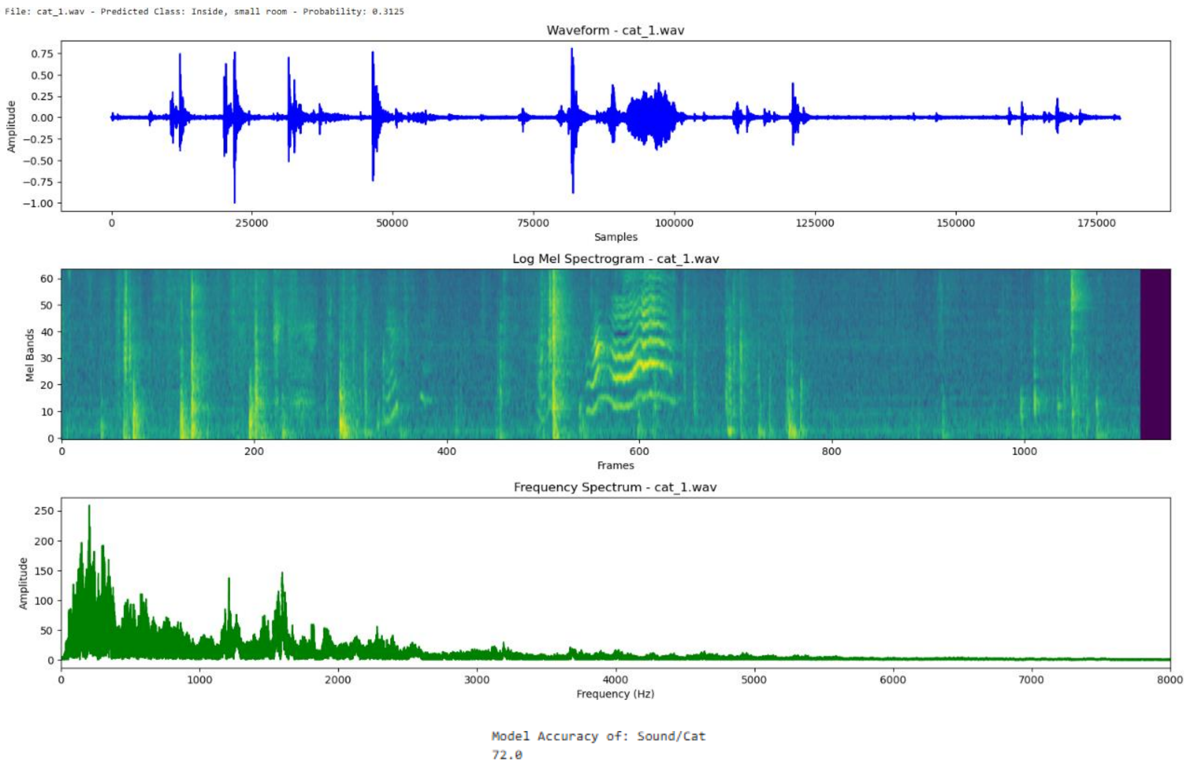


Figure 8: Example and base result of Cat.

As can be seen from the graph, unlike lion, cat has smaller loudness. Moreover, the sound only focusses on a few small parts of the audio. Moving to the Frequency spectrum, the distribution of sound frequency spreads more than lions with more patterns over 3000 Hz.

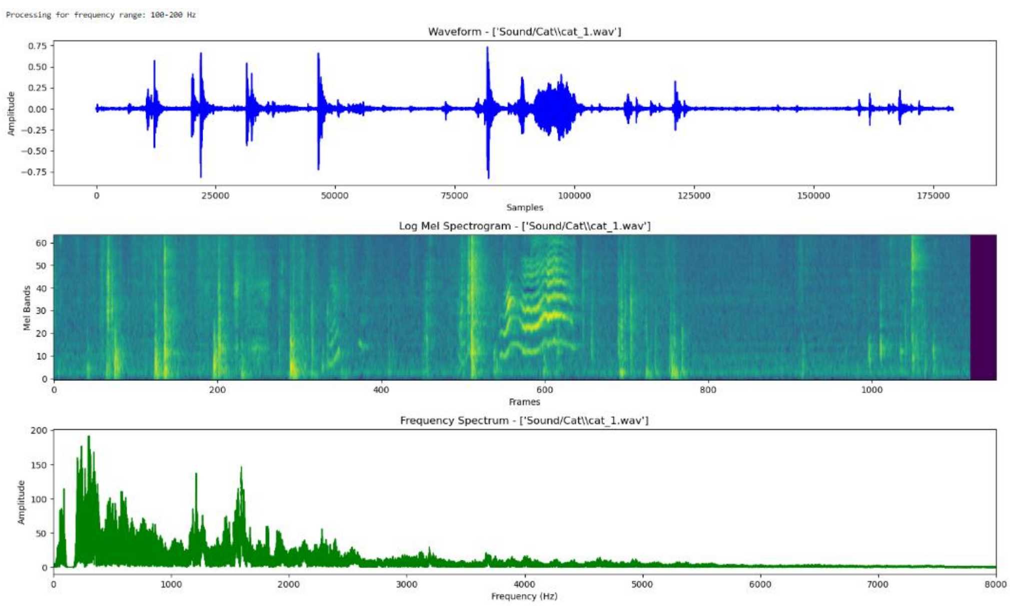


Figure 9: Audio result with Bandstop filter.

Using Bandstop filter, the removed frequency shows a significant change in loudness to nearly 200. However, there is almost no change in other frequency part. As a result, the only impact of bandstop filter for this animal is removing the most important pattern of the sound.

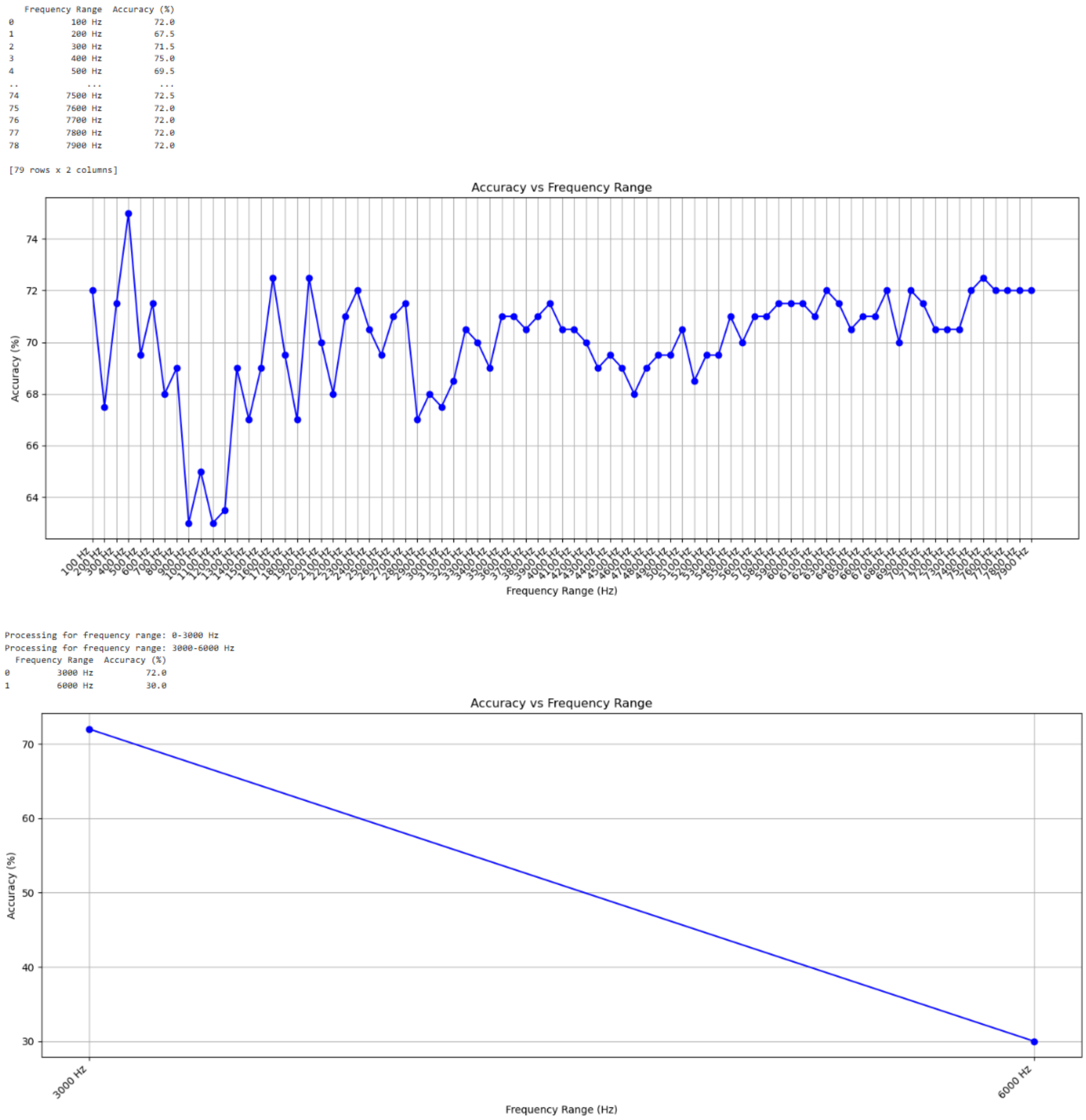


Figure 10: Result of using Bandstop Filter (top) and Bandpass Filter (bottom).

For this species, there are continuous ups and downs in accuracy in the whole process. Significantly, the frequency range from 1000 Hz to 1400 Hz plays a vital role in contributing to feature of cat. As can be seen in the graph, the accuracy decreases to 67% before increasing to 75%, followed by a huge drop to over 60% only in the first 1400 Hz. This indicates that while the 400 – 500 pattern is considered as noise data, the 1000 – 1400 part plays a vital role in the feature of cats. The result is further confirmed when the data is kept with the first 3000 Hz has significantly higher accuracy than for the next 3000 Hz. Finally, same to lion, the changes do not show much from 3000 Hz onward.

3) Dog:

- Desired class: Dog; Crunch; Domestic animals, pets; Animals.

Moving on to Dog, this is another domestic animal. However, unlike cat, dogs have low focuses on low frequency like lions but higher amplitude. With 100 files, I have inspected, and these are my conclusions. In the waveform visualization, dog audio contains short duration but high amplitude (around 0.5, higher than lion). However, in the frequency spectrum, dog concentrates mostly between 200 to nearly 2000 Hz. Therefore, removing the part between 100 to 200 Hz does not affect much the feature of dog. When using the Bandstop Filtering, while 1200 to 1500 and 1700 to 2000 are considered the noise of data, the range 700 Hz to 900 Hz is considered the most important feature of the data.

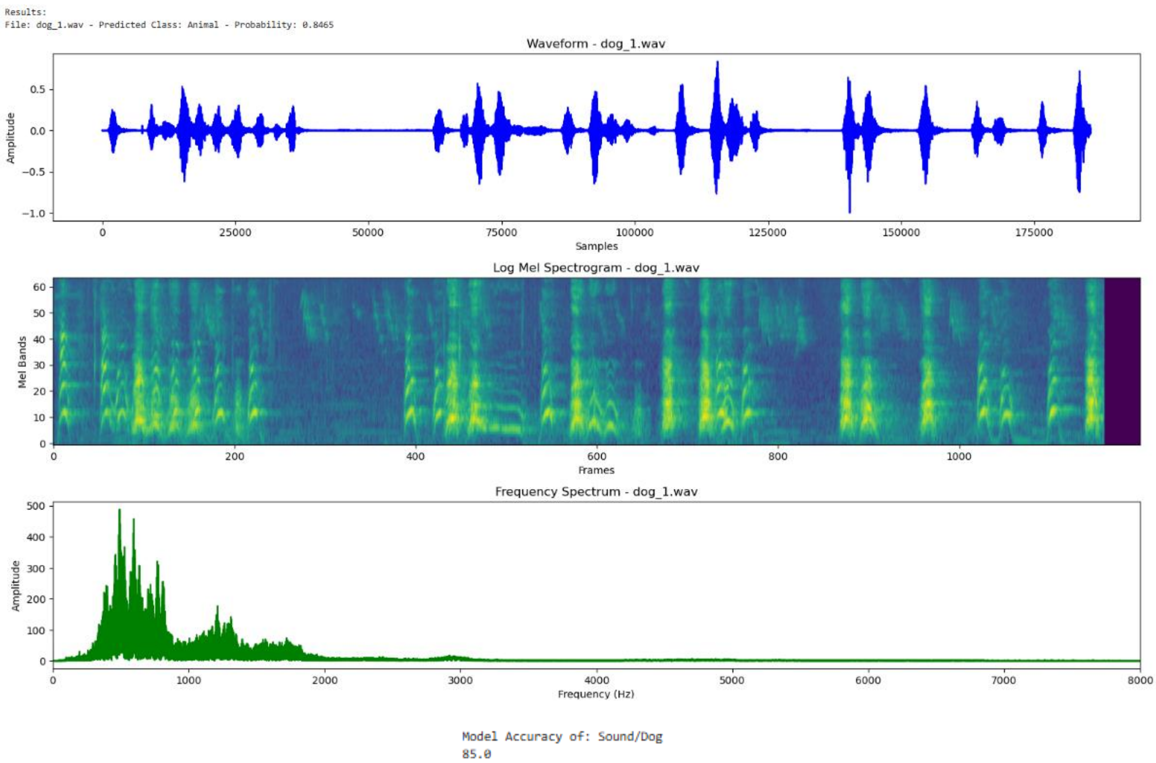
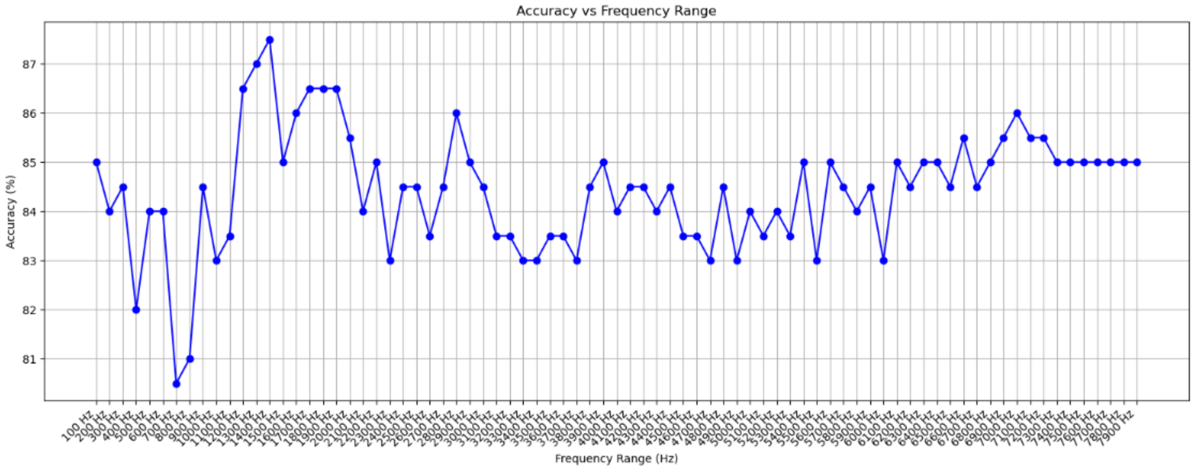


Figure 11: Example of Audio visualization of dog.

```
   Frequency Range  Accuracy (%)
0           100 Hz          85.0
1           200 Hz          84.0
2           300 Hz          84.5
3           400 Hz          82.0
4           500 Hz          84.0
..             ...           ...
74         7500 Hz          85.0
75         7600 Hz          85.0
76         7700 Hz          85.0
77         7800 Hz          85.0
78         7900 Hz          85.0

[79 rows x 2 columns]
```




```
Processing for frequency range: 0-3000 Hz
Processing for frequency range: 3000-6000 Hz
  Frequency Range  Accuracy (%)
0         3000 Hz          85.0
1         6000 Hz           6.0
```

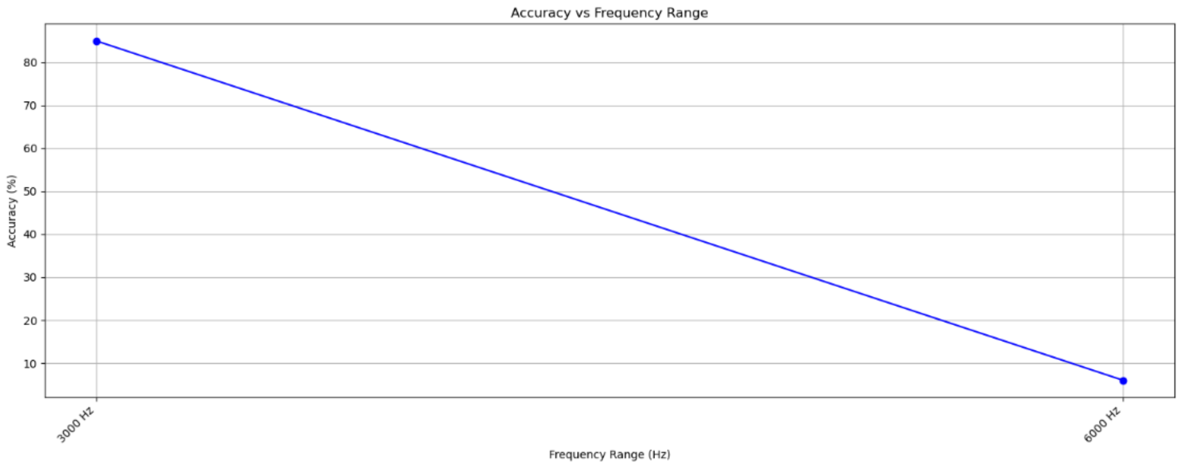


Figure 12: Result of using Bandstop Filter (top) and Bandpass Filter (bottom).

4) Koyun (Sheep):

- Desired class: Sheep, Animals.

For ruminants, sheep is my chosen among all of them. Using 40 samples, I have observed some interesting information. In waveform, the sound of sheep is a long with high frequency and amplitude. Specifically, while the amplitude reaches its peak at nearly 1, the frequency spreads from 1000 to 4000 Hz. Therefore, the Bandstop filter only affect the frequency above 1000. As a result, the accuracy drops significantly to 65% when removing the 1000 to 1100 Hz patterns. For other parts, there is not much change in accuracy from 4600 Hz onward.

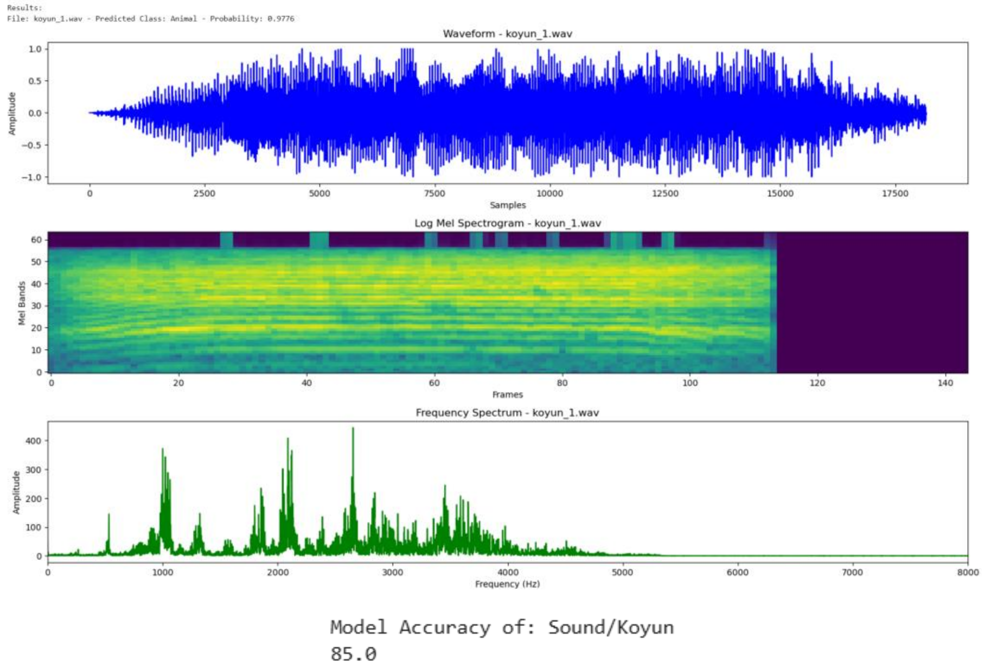


Figure 13: Example of Sheep.

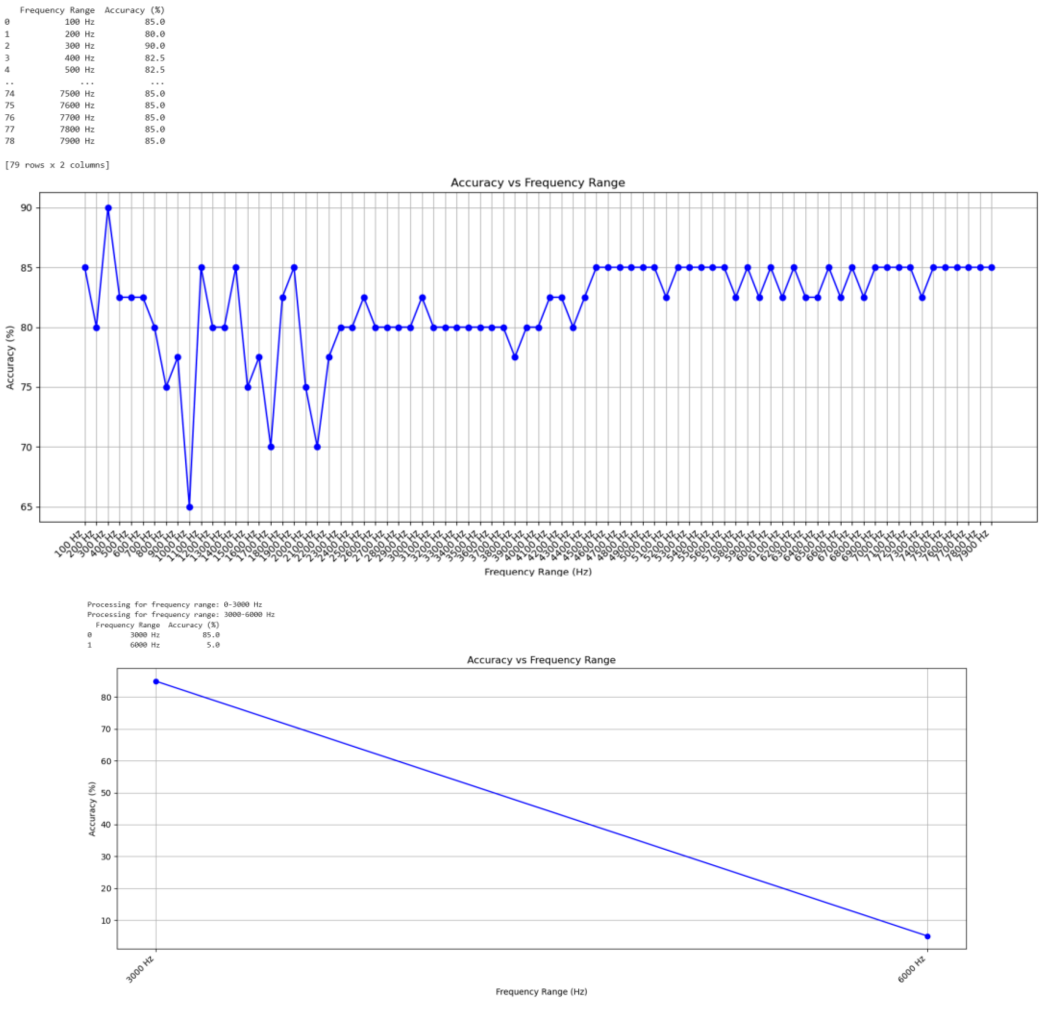


Figure 14: Result of using Bandstop and Bandpass.

5) Kurbaga (Frog):

- Desired class: Frog; Animal.

Moving on to the wild animals, Frog is one of the animals that has sound frequency focuses on medium but remains high amplitude. To digest deeper, while the amplitude keeps unchanged at nearly 1, the distribution of frequency only in 500 to 2000 Hz. Therefore, when using Bandstop Filter, the patterns between 100 to 200 Hz drops around 10% of accuracy. However, some small details like 1100 – 1200, 1300 -1400, 3500 – 3600, etc increases the accuracy to even 94%, making these are also important noise that needs to be concerns. Along with waveform visualization, the noise can be alternated with features. However, the accuracy still remains stable from 3000 onward.

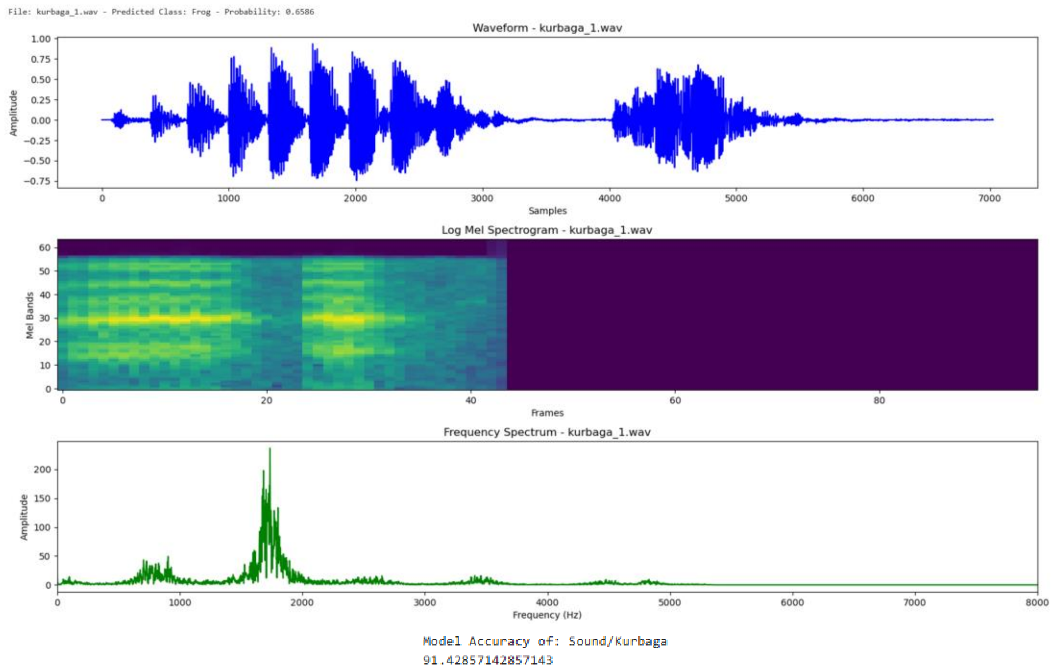
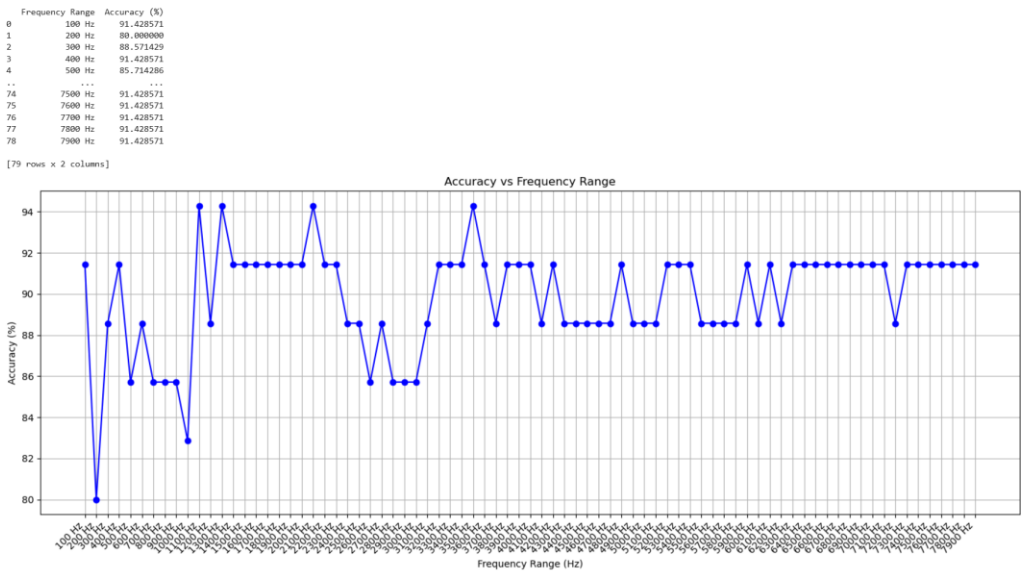


Figure 15: Example of Frog.



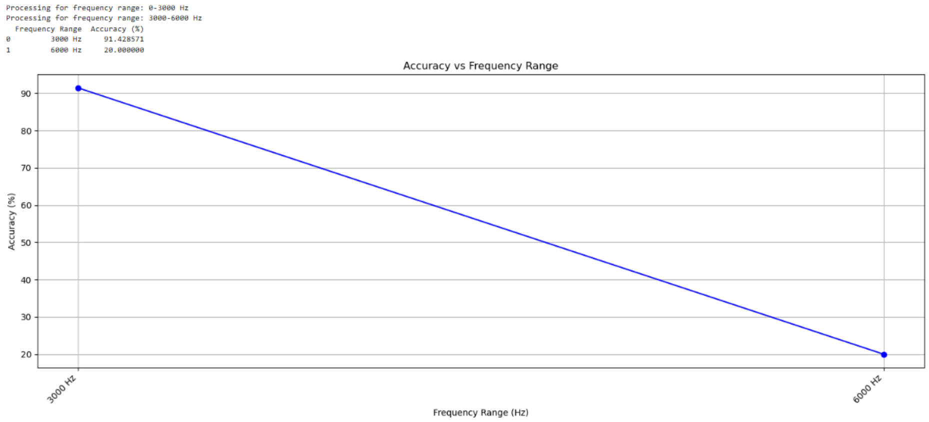


Figure 16: Bandstop and Bandpass result.

6) Result summary:

Below is my table displaying the result from all above:

Animals	Accuracy	Frequency range (Hz)	Maximum Amplitude	Feature Pattern(s) (Hz)	Noise Pattern(s) (Hz)	Bandstop affection	Bandpass Affection
Lion	88.89%	200 – 1200	350	300 – 400	800, 1100 - 1300	85% - 97%	-66%
Cat	72%	0 - 1500	190	900 - 1400	400 - 500	62% - 75%	-42%
Dog	85%	500 - 200	500	700 – 900	1200 – 1500	75% - 89%	-80%
Sheep	85%	1000 – 4000	400	1000 – 1100	200 - 300	65% - 90%	-80%
Frog	91.43%	500 – 2000	250	200 – 300	1000 – 1500	80% - 95%	-70%

E) Conclusions:

After conducting experiments in those audios of animal, I found that most of the animals have frequency range from 500 to 2000 Hz. However, some of them have feature patterns between 3000 and 4000 Hz. Furthermore, there are some important impacts of the Frequency Band Remover. Although Bandstop Filter can remove noise details and increase the accuracy of model, it also decreases dramatically the performance of the model when removing it feature patterns of animals. For Bandpass filter, it is not recommended to use when the feature pattern is not confirmed and should be used to verify the result from Bandpass. This is because removing the feature also increases the weight of irrelevant features.

F) Guide for Inspect the code and Reference:

1) Guide to test the code:

- Installed required library as listed in the code.
- Put the file and locate the file in the set-up part.
- Adjust the setting of frequency range for desired output.

2) Reference:

- Link 1.
- Link 2.
- Link 3.
- Link 4.
- Link 5.
- Data Link.