Elephant emotion detection using acoustic signals

Authors Names:
Arpudha T K – CB.EN.U4CCE22007
Ashmita D – CB.EN.U4CCE22009
Dhanu Raghavi S – CB.EN.U4CCE22018
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India

Abstract—**Elephant communication involves various vocalizations that convey various emotional states, social interactions, and environmental cues. This study explores the application of acoustic signal analysis for detecting and interpreting emotional states in elephants. Our approach involves gathering large-scale acoustic datasets from captive and wild elephant populations. This research project focuses on the application of machine learning techniques for emotion detection in audio signals. The study presents an integrated approach to signal processing, feature extraction, and classification. The proposed methodology aims to explore the effectiveness of Random Forest and Decision Tree models in classifying emotions based on audio features.**

Keywords—emotion detection, fast fourier transform, filtering, preprocessing, random forest, decision tree.

I. INTRODUCTION

This research project embarks on a comprehensive exploration of elephant communication, specifically focusing on the intricate vocalizations that serve as a key medium for conveying emotional states, social interactions, and responses to the environment. Elephants, renowned for their complex and intelligent social structures, exhibit a diverse range of vocal expressions that have been observed in both captive and wild populations. In an effort to deepen our understanding of these communication patterns, the study employs an innovative approach centred around acoustic signal analysis.

The primary objective is to discern and interpret the emotional state embedded within the tapestry of elephant vocalizations. To achieve this, the research methodology involves the collection of extensive acoustic dataset from diverse elephant populations. These dataset serve a foundation for multifaceted analytical process that integrates signal processing, feature extraction, and classification techniques.

Signal processing techniques such as fast fourier transform is crucial in enhancing the quality and clarity of the collected acoustic data, setting the stage for subsequent indepth analysis. Feature extraction then involves identifying and isolating relevant acoustic features that encapsulate the nuances of emotional expression in elephant vocalizations. Finally, the study employs machine learning techniques, specifically Random Forest and Decision Tree models, to categories and classify the identified emotional states based on the extracted features.

By combining advanced tools with the inherent richness of elephant vocalizations, this research not only contributes to our fundamental knowledge of elephant communication but also explores practical applications, such as emotion detection through machine learning. The potential implications, such as emotion detection through machine learning. The potential implications of this study extend beyond scientific inquiry, holding promise for informing conservation strategies, enhancing captive elephant welfare, and fostering a deeper connection with these remarkable and socially intricate beings.

The rest of the work is described as follows: Chapter II lists a few ongoing research works; Chapter III explains the inner details of the proposed method of using signal processing and machine learning techniques; Chapter IV illustrates the experimental setup and obtained results; Chapter V provides a few conclusions and finally Chapter VI includes the papers referred to in the project work.

II. RELATED WORK

In general, the assessment of emotions is a function performed by interconnected brain structures. While humnas can easily detect and express emotions through verbal and vocal communication, the same task is more challenging with animals due to their limited ability for verbal and vocal reporting.

Similar to humans, animals experience emotions such as happiness, anger, sadness, and fear [2]. Recognizing and capturing these emotions is crucial to prevent conflicts between humans and animals.

Examining specific animals such as elephants, researchers have focused on identifying emotional traits such as spreaded ears as an indication of fear in elephants[9]. Additionally, it has been observed that interpreting emotions from human faces does not universally apply to animals, despite some shared similarities among mammals[3],[7].

Recent research has incorporated sensors to capture the emotional state of animals, presenting innovative approaches

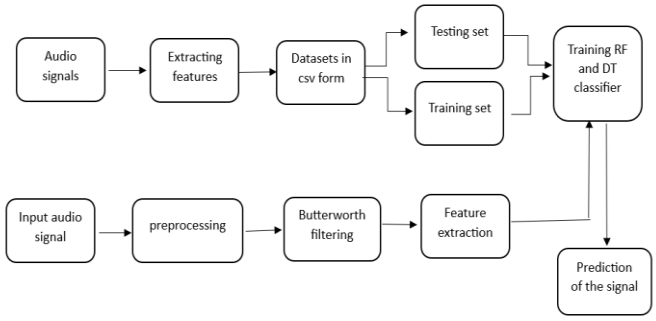
to understanding their emotions[6]. For example, a study by authors[5] employed cameras affixed to the ears of pigs to monitor their behaviour. More recently, Thangavel et al.[4] harnessed emotional intelligence in elephants by processing acoustic features on cloud servers.

However, a limited number of research works have explored the variety of elephant emotions by acoustic signal processing and machine learning techniques.

This article introduces a framework proposing emotion detection in elephants through signal processing and machine learning techniques. The aim is to use advanced technologies to better understand how elephants feel, filling gaps in our research.

III. Proposed Methodology

This section explains the methodology of elephant emotion detection using acoustic signals. It primarily is related to identifying the animal emotions by processing the acoustic signals obtained from the elephants using various signal processing techniques such as FFT, normalization, and Machine learning techniques such as Random Forests and Decision Trees.



A. *Data Preprocessing :*

- The input audio signal is read from a WAV file and a high-pass Butterworth filter is applied to the audio data.
- The FFT is a powerful tool for analyzing the frequency content of signals. FFT is employed for spectral analysis and feature extraction from audio signals.

B. *Filtering :*

- Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. It attenuates frequencies beyond the cutoff frequency without introducing ripples or sharp transitions in the passband.
- The transfer function H(s) of an nth–order Butterworth filter in the continuous time domain is given by :

$$H(s) = \frac{1}{\sqrt{1+(\frac{s}{\omega_c})^{2n}}}$$

Where :

s is the complex frequency variable

ω_c is the cutoff angular frequency

n is the filter order

- This filtering technique allows the higher-frequency components to pass through while attenuating lower–frequency components, effectively isolating high–frequency content in the audio signal.

C. *Feature extraction:*

- The feature extraction process involves computing MFCCs, spectral contrast, and zero-crossing rate from the audio signal.
- Features are then processed (padded and normalized) before being used as input for machine learning models for classification. Feature extraction is a crucial step in transforming raw audio data into a format suitable for machine-learning tasks
- MFCCs are coefficients representing the short-term power spectrum of a sound signal, capturing information about its spectral content.
- Spectral contrast measures the difference in amplitude between peaks and valleys in the spectrum and provides information about the distribution of energy in different frequency bands.
- Zero-crossing rate is a measure of how often the signal changes its sign, providing information about the noisiness or percussiveness of the signal.

D. *Normalisation:*

- The normalization technique used is Max Abs Normalisation , it is a technique used to scale and normalize data by dividing each data point by the maximum absolute value within the dataset.
- The formula used for computing Max Abs Normalization of single data point x is :

$$x_{normallised} = \frac{x}{\max(|x|)}$$

Where, max(|x|) is the maximum absolute value within the dataset.

- The result is that all data points are now scaled such that the maximum absolute value in the dataset becomes 1, and other values are scaled proportionally.

E. Classifier training :

- Classifier training is a crucial process in machine learning, aiming to equip a model with the ability to discern patterns and relationships within a labeled dataset.
- Beginning with input data containing features and corresponding labels, the process involves selecting an appropriate classifier algorithm, such as decision trees or Random forests.
- The training set is a subset of the data set used to train the machine learning model. It consists of input features and their corresponding labels or target values.
- The primary purpose of the training set is to learn patterns, relationships, and dependencies in the data. The model uses the input features to make predictions, and during training, it adjusts its internal parameters to minimize the difference between its predictions and actual labels.
- The testing set is a separate subset of the dataset that is not used during the training phase. It contains input features and corresponding labels or target values, similar to the training set.
- Once the model is trained on the training set, it is tested on the testing set to assess how well it generalizes to examples it has not encountered during training.
- Evaluation metrics such as accuracy, precision, recall, or F1 score are calculated on the testing set to quantify the model's performance. These metrics provide insights into how well the model is likely to perform on new, unseen data.

F. Prediction of signal :

- The output of the machine learning model is the predicted class or label for the new example, such as an audio signal.
- For a Random Forest model, the final prediction is often determined by majority voting across multiple trees in the ensemble.
- For a Decision Tree model, the prediction is based on the leaf node reached during the traversal of the tree structure.
- The predicted class represents the model's assessment of the type or category to which the input belongs.
- This output serves as the model's response to the specific characteristics of the new data, guided by the patterns learned during training.

IV. RESULTS AND DISCUSSION

1. *Dataset Description :*

The dataset utilized in this study comprises a comprehensive collection of elephant vocalizations sourced from elephant voices, offering resources for training and evaluating emotion recognition models. To facilitate feature extraction, the raw data is meticulously converted into CSV files, enabling the extraction of essential features like MFCCs, Chroma, and spectral contrast from the acoustic signals. Each audio signal is labelled with its corresponding emotion category, forming the basis for supervised learning. The dataset is subsequently partitioned into distinct training and testing sets, enabling the machine-learning models to learn from a portion of the data and assess its performance on unseen examples.

2. *Performance metrics :*

- Precision - In the proposed model, precision is used as a metric in classification models to assess the accuracy of the positive predictions made by the model. The computation of precision was determined in below equation :

$$precision = \frac{TP}{TP + FP}$$

Where,

TP – True positives

FP – False positives

- Recall(Sensitivity or True Positive Rate) – The recall (R) metric is the measure of the ability of the classifier to capture all the possible positive instances. The formulation of recall was illustrated by the below equation :

$$R = \frac{TP}{TP+FN}$$

Where,

TP – True positives

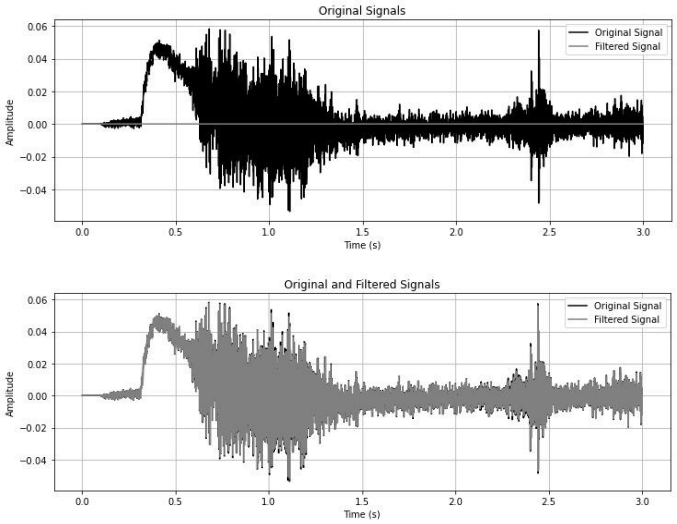
FN – False Negatives

- F1 Score – The harmonic average between precision and recall was used to define the F1 score. The F1 score is computed using the below equation :

$$F1 - Score = 2\ X\ \frac{Precision\ X\ Recall}{Precision+Recall}$$

- Support – The number of actual occurrences of the class in the specified dataset.

3. *Simulation results :*



Confusion matrix – The confusion matrix gives the performance of the machine learning model.

Confusion Matrix (Random Forest):

[[1 0 0 0 0 0]
[0 2 0 0 0 0]
[1 0 0 0 0 0]
[0 0 0 3 0 0]
[0 0 0 0 1 1]
[0 0 0 0 2 0]]

Classification Report (Random Forest):

	precision	recall	F1score	support
Baroo	0.50	1.00	0.67	1
Blast	1.00	1.00	1.00	2
Cry rumble	0.00	0.00	0.00	1
Roar	1.00	1.00	1.00	3
Roar rumble	0.33	0.50	0.40	2
Trumpet	0.00	1.00	1.00	2

accuracy			0.64	11
Macro avg	0.47	0.58	0.51	11
Weighted avg	0.56	0.64	0.59	11

Confusion Matrix (Decision Tree):
[[1 0 0 0 0 0]
[0 2 0 0 0 0]
[1 0 0 0 0 0]
[0 2 0 1 0 0]
[2 0 0 0 0 0]
[0 0 0 0 0 2]]

Classification Report (Decision Tree):

	precision	recall	F1score	support
Baroo	0.25	1.00	0.40	1
Blast	0.50	1.00	0.67	2
Cry rumble	0.00	0.00	0.00	1
Roar	1.00	0.33	0.50	3
Roar rumble	0.00	0.00	0.00	2
Trumpet	1.00	1.00	1.00	2

accuracy			0.55	11
Macro avg	0.46	0.56	0.43	11
Weighted avg	0.57	0.55	0.48	11

The predicted class for the input audio (Decision Tree) is: blast

The predicted class for the input audio (Random Forest) is: blast

4. *Inferences:*

- While the project appears to be well-structured and functional, several challenges and considerations may have been encountered during its development.
- The success of machine learning models for emotion recognition heavily relies on the quality and quantity of the training data. Obtaining a diverse and sufficiently large dataset with accurately labeled emotional states can be challenging.
- Tuning the parameters and finding the right balance between preventing overfitting and maintaining model performance is a non-trivial task. Optimizing for scalability would be essential for handling a broader range of applications.

- Processing audio streams in real-time introduces additional challenges related to latency and computational efficiency.
- Rigorous testing and validation, especially when dealing with various audio formats, lengths, and qualities, are crucial.

V. Conclusion

This project focuses on the analysis of audio data, specifically in the context of emotion recognition from elephant acoustic signals. This demonstrates a comprehensive approach, encompassing various signal processing and feature extraction techniques to distill relevant information from audio signals. The project's modular structure, with functions for filtering, feature extraction, and CSV output, enhances code readability and maintainability. Furthermore, the project sets the stage for future exploration and enhancement. Possible directions include the incorporation of advanced machine learning models for emotion classification, integration with larger datasets to improve model generalization, and the exploration of real-time applications. As the field of audio signal processing and emotion recognition continues to evolve, this project serves as a valuable foundation for further research and development in the domain.

VI. References

[1] . Y. -J. Chiu, S. Benedict and M. Gerndt, "E2D2: Elephant Emotion and Distraction Detection Framework using Edge-Enabled YOLOv5 Deep Learning Algorithm," *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Bangalore, India, 2023, pp. 1-6, doi: 10.1109/InC457730.2023.10262860.

[2] Ferres K., Schloesser T., and Gloor, P. A., Predicting Dog Emotions Based on Posture Analysis Using DeepLabCut, In Future Internet, Vol.14, No. 4, pp.1–16, https://doi.org/10.3390/fi14040097, 2022.

[3] Lauren R Finke, Stelio P. Luna, Juliana T. Brondani, Yorgos Tzimiropoulos, John McDonagh, Mark J. Farnworth, Marcello Ruta & Daniel S.Mills, Geometric Morphometrics for the Study of Facial Expressions in Non-human Animals, Using the Domestic Cat as an Exemplar, in Scientific reports Journal, Vol. 9, No. 1, pp. 1–12, 2019.

[4] Thangavel, Surya and Shokkalingam, Chitra Selvi, The IoT based embedded system for the detection and discrimination of animals to avoid human–wildlife conflict, in Journal of Ambient Intelligence and Humanized Computing, Vol. 13, No. 6, pp. 3065–3081, 2022

[5] Pandey, Santosh and Kalwa, Upender and Kong, Taejoon and Guo, Baoqing and Gauger, Phillip C and Peters, David J and Yoon, Kyoung Jin, Behavioral Monitoring Tool for Pig Farmers: Ear Tag Sensors, Machine Intelligence, and Technology Adoption Roadmap, in Animals, Vol. 11, No. 9, pp. 1–12, 2021

[6] Stoeger AS, Baotic A. Operant control and call usage learning in African elephants. Philos Trans R Soc Lond B Biol Sci. 2021 Oct 25;376(1836):20200254. doi: 10.1098/rstb.2020.0254. Epub 2021 Sep 6. PMID: 34482733; PMCID: PMC8419571.

[7] African elephants address one another with individually specific calls Michael A. Pardo, Kurt Fristrup, David S. Lolchuragi, Joyce Poole, Petter Granli, Cynthia Moss, Iain Douglas-Hamilton, George Wittemyer bioRxiv 2023.08.25.554872; doi: https://doi.org/10.1101/2023.08.25.554872

APPENDIX

```
import os
import librosa
import numpy as np
import pandas as pd
from scipy.signal import butter, lfilter
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report,
    confusion_matrix
 from sklearn.model_selection import train_test_split
 import matplotlib.pyplot as plt
 def manual_fft(x):
 N = len(x)
 log2_N = int(np.ceil(np.log2(N)))
padded_size = 2 ** log2_N
x = np.pad(x, (0, padded_size - N), 'constant')
assert len(x) == padded_size, "Error in zero-padding"
for i in range(padded_size):
 j = int('{:0{width}b}'.format(i, width=log2_N)[::-1], 2)
if i < j:
x[i], x[j] = x[j], x[i]
for m in range(2, padded_size + 1, 2):
offset = padded_size // m
for start in range(0, padded_size, m):
butterfly(x, start, offset, m // 2, padded_size)
return x.astype(complex)
def butterfly(arr, start, offset, m, N):
end = start + m
even_part = arr[start:end]
if end < len(arr):
odd_part = arr[end:end+m] * np.exp(-2j * np.pi *
   np.arange(m)/m)
 arr[start:end] = even_part + odd_part
 arr[end:end+m] = even_part - odd_part
def highpass_butterworth_manual(data, cutoff, fs,
  order=5):
 nyquist = 0.5 * fs
 normal_cutoff = cutoff / nyquist
 x = np.linspace(0, order, order + 1)
 h = np.sinc(2 * normal_cutoff * (x - order / 2.0))
 h *= np.blackman(order + 1)
 h /= np.sum(h)
 filtered_data = np.convolve(data, h, mode='same')
 return filtered_data
def plot_signal(original_data, filtered_data, sample_rate,
    title):
time = np.arange(0, len(original_data)) / sample_rate
plt.figure(figsize=(12, 4))
plt.plot(time, original_data, label='Original Signal',
    color='black')
plt.plot(time, filtered_data, label='Filtered Signal',
    color='grey')
plt.title(title)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.legend()
plt.grid(True)
plt.show()
def plot_fft(data, sample_rate, title):
N = len(data)
 T = 1 / sample_rate
xf = np.linspace(0.0, 1.0/(2.0*T), N//2)
 yf = 2.0/N * np.abs(manual_fft(data)[:N//2]
plt.figure(figsize=(12, 4))
plt.plot(xf, yf)
plt.title(title)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.show()
def extract_features(audio, sample_rate, mfcc=True,
    Chroma=True, mel=True):
plot_signal(audio, np.zeros_like(audio), sample_rate,
```

```
    'Original Signals')
[67] cutoff_frequency = 50.0
[68] filtered_audio = highpass_butterworth_manual(audio,
    cutoff_frequency, sample_rate)
[69] plot_signal(audio, filtered_audio, sample_rate, 'Original
    and Filtered Signals')
[70] features = []
[71] if mfcc:
[72] mfccs = np.mean(librosa.feature.mfcc(y=filtered_audio,
    sr=sample_rate, n_mfcc=13), axis=1)
[73] features.extend(mfccs)
[74] if chroma:
[75]chroma= np.mean(librosa.feature.chroma_stft(y=
    filtered_audio, sr=sample_rate, axis=1)
[76] features.extend(chroma)
[77] if mel:
[78]mel = np.mean(librosa.feature.melspectrogram(y=filtered
    _audio, axis=1)
[79] features.extend(mel)
[80] return features
[81] def save_features_to_csv(features, output_csv):
[82] df = pd.DataFrame([features], columns=[f'feature_{i}'
for
    i in range(len(features))])
[83] df.to_csv(output_csv, index=False)

[84] def predict_audio_class(audio_features, classifier):
[85] return classifier.predict([audio_features])
[86] audio_file_path = r"D:\# SEM 3 PPT\DSP\# PROJ\wav
    files\inputs\input.wav"
[87] output_csv_path = 'audio_features.csv'
[88]audio, sample_rate = librosa.load(audio_file_path,
    res_type='kaiser_fast', duration=3)
[89] audio_features = extract_features(audio, sample_rate)
[90] save_features_to_csv(audio_features, output_csv_path)
[91] data = pd.read_csv(r"D:\# SEM 3 PPT\DSP\#
    PROJ\emotion_dataset_2.csv")
[92] X = data.drop('label', axis=1)
[93] y = data['label']
[94] X = np.array(X) / np.max(np.abs(X))
[95] X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)
[96]clf_random_forest=RandomForestClassifier(n_estimators
    =100, random_state=42)
[97] clf_random_forest.fit(X_train, y_train)
[98]clf_decision_tree=DecisionTreeClassifier(random_
    state=42)
[99] clf_decision_tree.fit(X_train, y_train)
[100] predicted_class_random_forest =predict_audio_class(
    audio_features, clf_random_forest)
[101]y_pred_random_forest = clf_random_forest.predict(
    X_test)
[102] print("\nConfusion Matrix (Random Forest):")
[103] print(confusion_matrix(y_test, y_pred_random_forest))
[104] print("\nClassification Report (Random Forest):")
[105] print(classification_report(y_test,
    y_pred_random_forest))
[106]predicted_class_decision_tree=predict_audio_class(audio
    _features, clf_decision_tree)
[107] y_pred_decision_tree = clf_decision_tree.predict(X_test)
[108] print("\nConfusion Matrix (Decision Tree):")
[109] print(confusion_matrix(y_test, y_pred_decision_tree))
[110] print("\nClassification Report (Decision Tree):")
[111] print(classification_report(y_test, y_pred_decision_tree))
[112] print(f"The predicted class for the input audio (Decision
    Tree) is: {predicted_class_decision_tree[0]}")
[113] print(f"The predicted class for the input audio (Random
    Forest) is: {predicted_class_random_forest[0]}")
```