ELEPHANT RUMBLE DETECTION REPORT

PREPARED BY

Yash Sharma

EXECUTIVE SUMMARY

The rise of technological advancements and increasing population are associated with more human indulgence with deforestation. This leads the wild animals to frequently visit human habitations. In rural areas these animals come in search of food and they harm the villagers, destroying crops and properties. Not only humans are affected but also the innocent mammals who just came in order to search for food.

The villagers try to drive them off by firing crackers, beating drums and other various ways. However they start driving them off when they are attacked. They do not have any sort of prior information of the advancing elephants.

In this report we discuss a deep learning model which can automatically detect the presence of elephants by detecting the low frequency sound produced by them also called ‘rumbles’.

Keywords Used:

Rumbles, GFCC, MFCC and CNN

A Brief Overview of Proposed Techniques

The most widely used technique to classify between different sounds is using the Mel Frequency Cepstrum Coefficient (MFCC). MFCCs allow us to extract the linguistic content and discard all the other stuff which carries information like background noise. In more formal terms the job of MFCCs is to determine the shape of vocal tract which could give us an accurate representation of the phoneme being produced, this shape determines what sound comes out and is regarded as an important feature to distinguish between different sounds.

The problem with MFCCs is that this methodology is built to distinguish between human sounds specifically using the knowledge of how human cochlea works and perceives sound. We need to improvise over current techniques as we are dealing with elephants. They perceive frequency on a logarithmic scale along the cochlea and the relationship between parameters is modeled differently from what humans have.

The approach proposed in literature for rumble detection is using the Greenwood Frequency Cepstral Coefficients(GFCCs). GFCCs essentially work the same as MFCCs does but there is a fundamental difference, GFCCs use the Greenwood scale to model frequencies while the MFCCs use the Mel Scale.

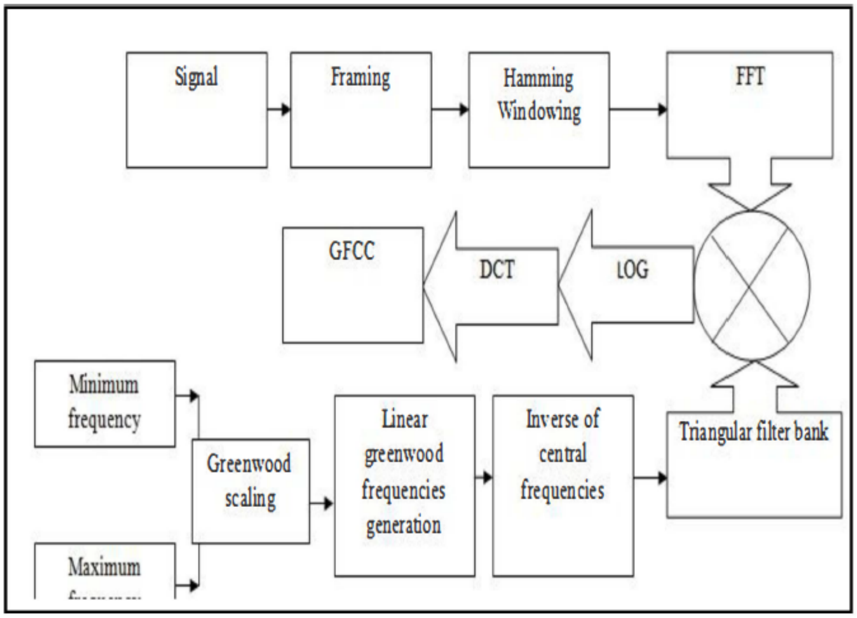
Proposed Method and Methodology

The data for Rumble files have a sampling frequency of 8000 Hz. Since most of the information lies within 500Hz the sound files were re-sampled with a sample rate of 2000Hz. The hearing range of elephants lie from 10Hz to 10000Hz. 30 filters have been used in a range of 10Hz to 500Hz which corresponds to frequency range of rumble to extract GFCC. Out of 30 GFCC coefficients 18 have been used as cepstral features to represent the sound.

The original audio files were composed both of noise and the rumbles. The audio files are divided into small fragments of 4 seconds each. If a rumble was less than 4 seconds it was padded with extra zero values and if it was bigger than 4 seconds it was trimmed. This was done in order to make all the files uniform so as to feed in a neural network.

Analysis window length of 75ms was used with a window step(the step between successive windows in seconds) of 25 ms. Preemphasis and lifter parameters were set to zero which implies none of these methods were applied. Hamming window function was used, hamming tapers off values at the high frequency for minimal discontinuity.

The figure below shows all the steps involved in GFCC extraction.



Model Overview and Results

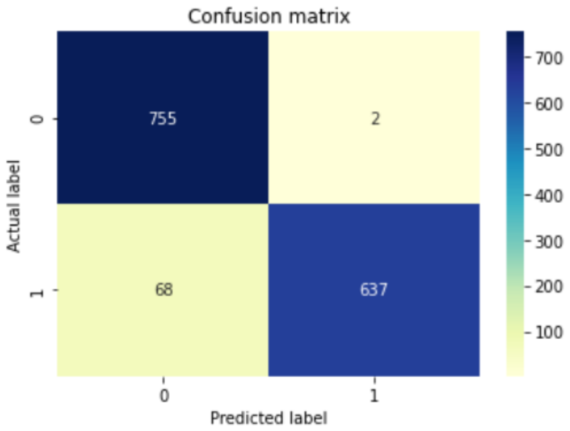
First, a dense model was built using the same parameters as mentioned above which didn’t give satisfactory results. Following this a Convolutional Neural Network was built with better accuracy and results. We’ll discuss the CNN model and its results in the upcoming section.

Model Architecture:

The CNN model is based on the sequential API of the tensorflow library. First layer is Conv2D layer with a size of 64 output space in other words 64 filters with a kernel size of 3 followed by a batch normalisation layer, an activation layer with activation function as ‘relu’ and a max pooling layer of pool size 2. These 4 layers are again applied but this time the number of filters is reduced to 32. Lastly the model features are flattened and a batch normalised dense layer with 32 parameters is added. Finally to classify between the two classes a final dense layer with a single neuron is added. Adam optimiser is used for optimization of gradient descent.

Results:

1. Model Accuracy on training set: 99.63%
2. Model Accuracy on validation set: 93.22%
3. Model Accuracy on test set: 94.39%
4. Confusion Matrix:



Label 1 denotes a rumble and 0 denotes noises

5. Precision: 0.96 (percentage of samples which were +ve and were correctly classified)
6. Recall: 0.95 (percentage of samples which were -ve and were correctly classified)

Note that the false negatives are 68 in number over a test set of size 1462 samples. In order to decrease the false negative count classification point(threshold) of sigmoid was changed from 0.5 to 0.45 without affecting other values.

Furthermore, another reason for the number of false negatives could be wrong labelling of the rumble files.