DEEP AUDIO CLASSIFIER

R GNANADEEP

K VS S K LOKESH

TEJA SRI HARSHA CH

INTRODUCTION

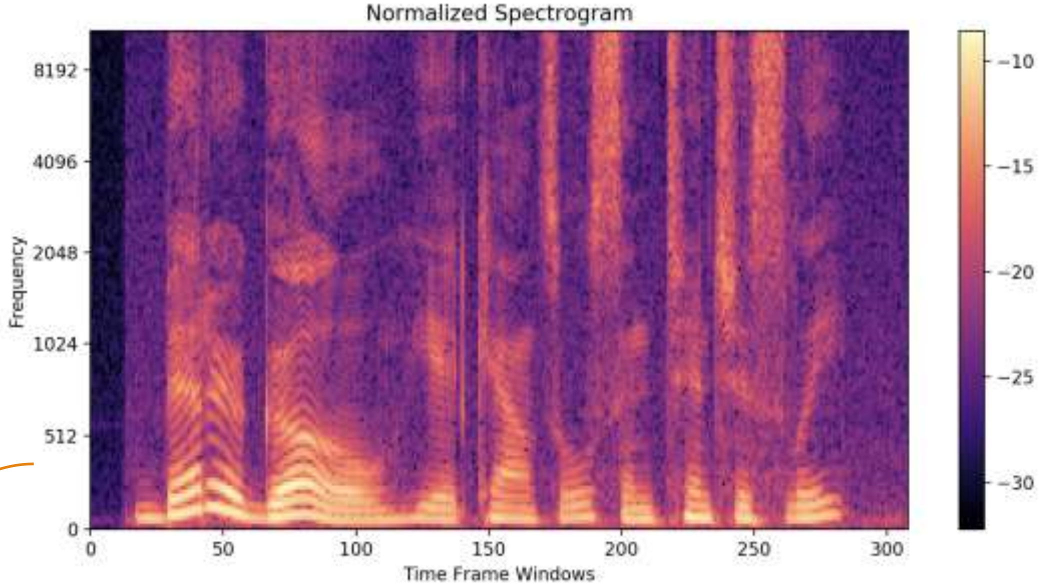
- An innovative system to automatically detect capuchin bird sounds in forest audio recordings using deep learning.
- This method employs Convolutional Neural Networks (CNNs) to analyze spectrogram representations of the audio, which capture both time and frequency data crucial for identifying these bird vocalizations amidst forest noise.
- By automating the identification process, our system streamlines efforts to track capuchin birds in forests and also has practical applications in wildlife management and environmental conservation.

STEPS

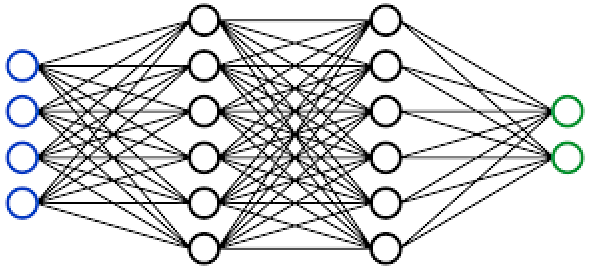
1) Convert Audio data to wave forms
2) Transform the wave form to spectrogram
3) Build the Convolutional Neural network
4) Classify Capuchin bird calls

Capuchin Bird

a)Audio data to waveforms



b)Wave forms to spectrogram data



c)Build Neural network and classify audio

Dataset Description

Dataset Name - Z by HP Unlocked Challenge 3 – Signal Processing

Dataset Link - https://www.kaggle.com/datasets/kenjee/z-by-hp-unlocked-challenge-3-signal-processing

Contains 3 folders

- Forest Recordings(raw audio from forest)
- Parsed_Capuchinbird_Clips(clips that contain Capuchin bird calls)
- Parsed_Not_Capuchinbird_clips(other animals and bird sounds in the forest)

(27% true labels, 73% false labels)

Data Preprocessing

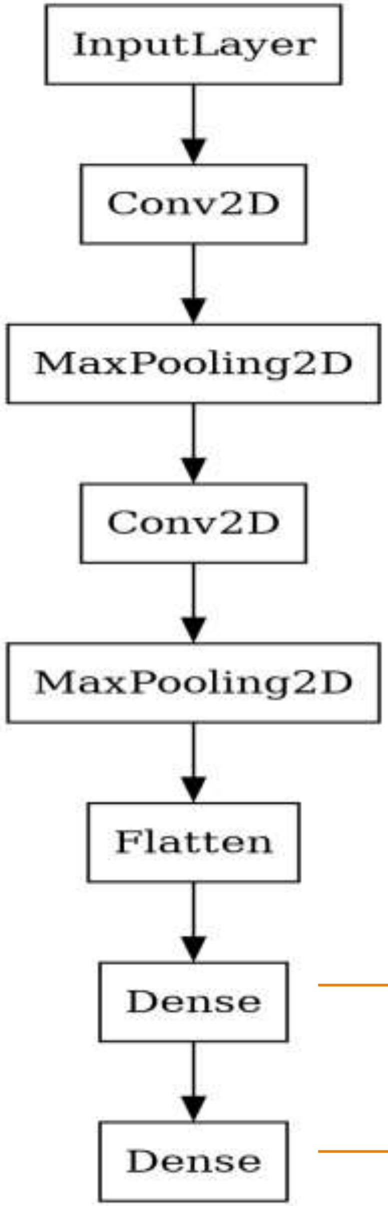
Down Sampling:

- Our initial dataset has a sampling frequency of 44100 Hz which is very high. So we need to down sample the audio for training. The audio is being down sampled to 16000Hz

Spectrogram:

- The loaded waveform is then trimmed or padded to a length of 48000 samples (which corresponds to 3 seconds of audio assuming a sample rate of 16 kHz).
- If the waveform is shorter than 48000 samples, it pads the waveform with zeros at the beginning to make it 48000 samples long. This is done using TensorFlow's tf.zeros() function.
- The function then computes the Short-Time Fourier Transform (STFT) of the audio waveform using TensorFlow's tf.signal.stft() function.

CNN Model Diagram



Model Architecture

Convolutional layer with 16 filters, each kernel with size of (3,3) and Max pooling layer of kernel size (2,2).

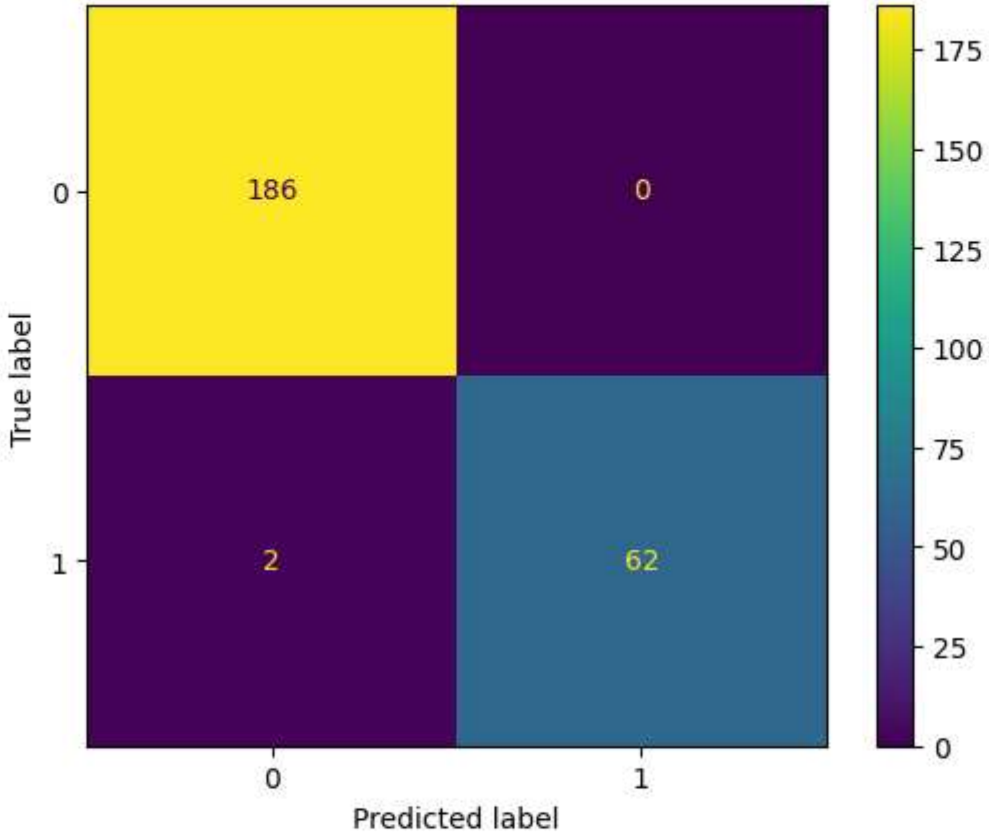
Convolutional layer with 16 filters, each kernel with size of (3,3) and Max pooling layer of kernel size (2,2).

Dense layer with 128 units and ReLu activation function

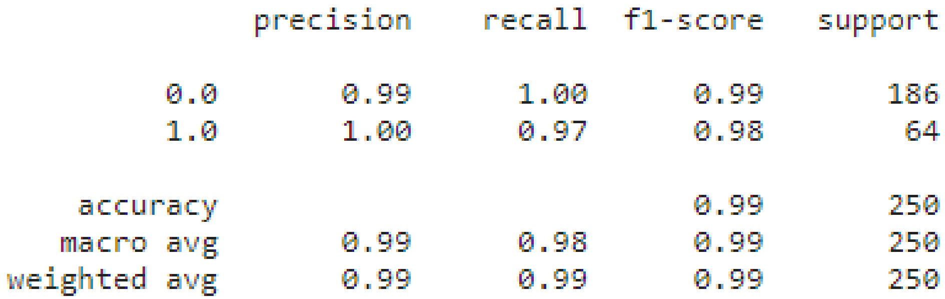
Output layer with sigmoid activation function

Metrics

Confusion matrix



Classification report



```
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99       186
         1.0       1.00      0.97      0.98        64

    accuracy                           0.99       250
   macro avg       0.99      0.98      0.99       250
weighted avg       0.99      0.99      0.99       250
```

Thank You