# Wild-ID: Real-Time Acoustic Classification of Different Wildlife Species

## CONTACT INFO:


- **Name:** Guneesh Gupta
- **Branch:** Electrical Engineering
- **Institute ID:** guneesh_g@ee.iitr.ac.in
- **Enrollment:** 25115060
- **Contact No:** 9815318074


## INTRODUCTION:

In this project, I aim to develop **Wild-ID**, a deep learning-based bioacoustic classification system capable of identifying wildlife species (such as insects, amphibians, and nocturnal mammals) from the sounds they make.

The system uses a **Signal-to-Image** pipeline, converting environmental audio into **Per-Channel Energy Normalized (PCEN)** spectrograms. The spectrograms are processed by a **Convolutional Recurrent Neural Network (CRNN)** to capture both the spectral texture (pitch) and temporal rhythm (tempo) of animal calls, distinguishing species even in noisy outdoor environments.

- **Data:** I will use the **ESC-50** dataset (Piczak, 2015) for baseline training and make a specific subset of "cryptic" sounds (cicadas, frogs, crickets) from the **Xeno-canto** repository.
- **Preprocessing:** Using the **Librosa** library, raw audio of animals will be converted into **Mel-Spectrograms** and further enhanced using **PCEN** (Wang et al., 2017) to suppress background wind noise and automatically balance the loudness.
- **Model:** A **CRNN** architecture (Cakir et al., 2017) will be implemented using PyTorch.The **CNN** identifies the unique shape of the sound, while the Bi-directional **LSTM** analyzes its rhythm and pattern over time.
- **Augmentation:** To prevent overfitting on limited bioacoustic data, I will implement **SpecAugment** (Park et al., 2019), applying time and frequency masking during training.

## MOTIVATION

I got motivated on this idea by **Shazam**. We can literally identify any song using it in 3 seconds, so why not use it for identifying insect and animal voices during **hiking or trekking**? It would be incredibly helpful to the hikers to differentiate between potential threats and benign sounds. The system is also applicable in **residential settings**, allowing homeowners to identify **unfamiliar wildlife calls** in their gardens or homes.
Additionally, I gained motivation from a research paper titled "Audio Detection of Chainsaws in Forests" [6].

## TIMELINE

- **Week 1 (Dec 5 – Dec 13)**
    - I will understand how to **read spectrograms** properly. I will study the **librosa** library for audio processing. I will start by **mapping the PCEN mathematical model to the PyTorch codebase**. I will study the specific PyTorch functions needed to implement the *PCEN* equation from Wang et al., effectively translating my plan into executable code.
    - I will download the **ESC-50** collection and sort the audio files into clear folders. Before writing code, I will manually listen to the files and analyse the differences in them
    - I will be translating my algorithmic understanding into **PyTorch** syntax. I will focus specifically on mastering **Tensor operations** and work on a custom **Dataset Class** to handle audio files.
    - I will develop a **prototype notebook** to experiment with loading a single `.wav` file and attempting to convert it into a **PCEN Spectrogram**. It will be like a test case.

- **Week 2 (Dec 14 – Dec 21):**
    - I must verify the baseline data. I will learn and write scripts to convert raw audio into standard **Log-Mel Spectrograms**.
    - Once the Mel-Spectrograms are generated, I will implement **Per-Channel Energy Normalization (PCEN)** on top of them. I will experiment with different bias/gain parameters to demonstrate how PCEN "cleans" the Mel-Spectrogram by suppressing background wind noise.

**22nd December: Mid-Term Evaluation**: **Data Analysis Report** containing a "Raw Mel-Spectrogram vs. PCEN-Enhanced Spectrogram." This will clear the difference between them to the judges and ensure that my data pipeline is ready and can be coded further

- **Week 3 (Dec 23 – Dec 29):**
    - I will translate the research paper diagrams into PyTorch code, constructing the **Convolutional layers** and **LSTM layers**.
    - I will make the **Training Loop** and feed in the processed dataset. I will train it to establish a **baseline accuracy**, verifying that the CRNN is actually learning from the PCEN features

- **Week 4 (Dec 30 – Jan 4):**
    - To prevent the model from memorizing the limited dataset (overfitting), I will integrate **SpecAugment** into the training pipeline. I will implement time and frequency masking to force the model to learn various typical features even when parts of the audio are missing.
    - If time permits, I will generate a Confusion Matrix to visualize exactly where the model fails. I will use this data to fine-tune the final model weights.
    - I will build the final **Inference Engine**. I will implement a **sliding window logic** that allows the model to scan continuous audio streams (simulating a real-world microphone feed) rather than just pre-cut clips.

## REFERENCES:


**[1] The Dataset:**

- **Citation:** Picza, K. J. (2015). **ESC: Dataset for Environmental Sound Classification.** *Proceedings of the 23rd ACM International Conference on Multimedia.*
- **Source:** https://github.com/karolpiczak/ESC-50

**[2] The Preprocessing (PCEN):**

- **Citation:** Wang, Y., et al. (2017). **Trainable Frontend For Robust and Far-Field Keyword Spotting.** *Google Research, ICASSP 2017.*
- **Source:** https://arxiv.org/abs/1607.05666

**[3] The Architecture (CRNN):**

- **Citation:** Cakir, E., et al. (2017). **Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection.** *IEEE Transactions on Audio, Speech, and Language Processing.*
- **Source:** https://arxiv.org/abs/1702.06286

**[4] The Augmentation (SpecAugment):**

- **Citation:** Park, D. S., et al. (2019). **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.** *Google Brain, Interspeech 2019.*
- **Source:** https://arxiv.org/abs/1904.08779

**[5] General Methodology:**

- **Citation:** Hershey, S., et al. (2017). **CNN Architectures for Large-Scale Audio Classification.** *Google Research, ICASSP 2017.*
- **Source:** https://arxiv.org/abs/1609.09430

**[6] Motivation:**

- **Citation:** R. White *et al.*, "Real-Time Audio Detection of Chainsaws in Forests using Convolutional Neural Networks," *Rainforest Connection, White Paper*, 2018.
- **Source:** https://rfcx.org/

## ABOUT ME:

I am a first-year B.Tech student in Electrical Engineering. I have a strong background in C++, and I know the basics of Python as well. I am familiar with NumPy and Matplotlib. I have a great interest in mathematics, physics, and logic. I have intuition-based knowledge regarding neural networks, and I love to discover more and more about them. I have also started following a framed coursework on Coursera regarding Deep learning and Machine Learning.

I am fascinated by this project as it includes the concept of "Signal-to-Image" processing—taking sound waves, converting them into visual Spectrograms, and teaching a machine to actually "see" a sound.

I see **Wild-ID** not just as a project, but as the foundational step in building my career in Machine Learning, giving me the practical, hands-on experience I need to grow from a beginner to a capable AI specialist.