Project Vision: Wildlife Signal Detection & Species Recognition System

**Core Idea**

Use your enhanced multi-component model to extract faint animal movements or sounds from noisy environmental data — and classify the detected signal into known species.

### **Step-by-Step Build Plan**

#### ****Phase 1: Signal Detection Engine****

* **Input:** Raw data from motion sensors, thermal cams, infrared mics, or audio traps
* **Your Model's Role:** Extract modulated or structured signals (footsteps, bird flutters, vocalizations)
* **Add-on:** Thresholding and residual filtering to ensure no false positives from wind, rain, or leaves

#### ****Phase 2: Lightweight Classifier****

* **Input:** Cleaned signal output from your model
* **AI Add-on:** A small neural net (e.g., MobileNet, or CNN+LSTM hybrid) trained to classify species — birds, monkeys, etc.
* **Training Data:** Public audio/video datasets like BirdCLEF, EcoSound, or visual thermal clips

#### ****Phase 3: Visualization App (MVP)****

* **Tool:** Build with Streamlit or Gradio
* **Features:**
  + Upload raw audio or video
  + Run your model in-browser
  + Show extracted signal regions
  + Display classification output (species + confidence)
  + Optional map overlay with timestamps

### **Stretch Goals**

* Add **real-time alerts** via email or text
* Sync detections with GPS for tracking
* Build a **dashboard** to monitor trends over time (migration, population, intrusions)

### **What You Can Build Now (Next 2–3 Weeks)**

|  |  |  |
| --- | --- | --- |
| **Task** | **Tool** | **Time Estimate** |
| Build signal detection backend | Your model + NumPy/SciPy | 3–5 days |
| Create species classifier | PyTorch or TensorFlow | 1 week |
| Frontend interface (MVP) | Streamlit/Gradio | 2–3 days |
| Data sourcing & labeling | BirdCLEF, Macaulay Library | ongoing |

### **Reliable Dataset Sources for Wildlife Signal Detection**

#### ****1. BirdCLEF (Xeno-Canto + LifeCLEF)****

* **Host:** LifeCLEF Challenge / Zenodo
* **Download:** Available via competition or direct access links from prior years
* **Content:** Labeled bird audio recordings (species + metadata)
* **Formats:** WAV/MP3, metadata in CSV

#### ****2. Macaulay Library (Cornell Lab of Ornithology)****

* **Link:** <https://www.macaulaylibrary.org>
* **Use:** Search by species, location, or contributor
* **Download Access:** Requires free account for bulk access
* **Note:** Excellent source for natural field recordings

#### ****3. EcoSound / DCASE Challenge****

* **Link:** https://dcase.community/challenge2021/task-urban-sound-tagging
* **Use:** Annotated environmental audio — including birds, insects, urban sounds
* **Format:** Suitable for training your classifier alongside your model

I’ll now generate a **PDF roadmap** that includes:

* Project architecture
* Dataset links
* MVP features
* Tools to use for each phase

## MVP GOAL

* Detect structured biological signals (bird calls, movements) in noisy audio
* Classify the species from those signals
* Show it all in a simple interactive UI

wildlife\_detector/

├── app/ # Frontend interface (Streamlit or Gradio)

│ ├── main.py # Launch script

│ └── ui\_helpers.py # UI layout, file upload, audio playback

│

├── core/ # Signal decomposition + model integration

│ ├── signal\_model.py # Your enhanced multi-component model

│ ├── preprocess.py # Resampling, normalization, windowing

│ └── postprocess.py # Residual filtering, component extraction

│

├── classifier/ # AI model for species recognition

│ ├── train\_classifier.py # CNN or CNN+LSTM training

│ ├── predict.py # Classify extracted components

│ └── model/ # Saved models (e.g., .pt or .h5)

│

├── data/ # Local or downloaded dataset folders

│ ├── raw/ # Original WAV/MP3 files

│ └── processed/ # Numpy or spectrogram-ready formats

│

├── utils/ # Logging, visualization, helpers

│ ├── logger.py

│ └── visualization.py

│

├── requirements.txt # Dependencies (torch, librosa, streamlit, etc.)

└── README.md # Project intro & setup instructions

## DATASETS TO USE

### 🎤 **BirdCLEF 2023**

* 🔗 https://www.imageclef.org/lifeclef/2023/bird
* ✅ Use for training/testing the species classifier
* 💡 Supports metadata like location, time, and species
* Format: .wav audio clips with CSV annotations

### 🐦 **Macaulay Library (Cornell)**

* 🔗 <https://www.macaulaylibrary.org>
* ✅ Use for additional testing or transfer learning
* ✳️ Best for diverse environments and natural variability

### 🌆 **EcoSound/DCASE Urban Sounds**

* 🔗 https://dcase.community/challenge2021/task-urban-sound-tagging
* ✅ Helps model generalization in noisy backgrounds (cars, wind, etc.)
* Format: .wav, .json annotations with multi-label tags

## 🛠️ Tech Stack (Recommended)

| **Task** | **Tool/Lib** |
| --- | --- |
| Audio loading/processing | librosa, scipy, soundfile |
| Signal modeling | numpy, your model logic in signal\_model.py |
| ML training | PyTorch or TensorFlow |
| Visualization | matplotlib, streamlit, plotly |
| Deployment | Gradio, Streamlit, or Docker for packaging |