BirdShazam: Bird Species Classification Framework Using Audio Recordings

GROUP 36

Problem statement

Core Problem:

Traditional bird species identification methods rely heavily on human expertise—requiring manual field surveys, sound interpretation, and expert sonogram analysis. These approaches are:

- Time-consuming
- Geographically limited
- Subjective and inconsistent
- Inefficient for large-scale biodiversity monitoring

Why It Matters:

Birds are early indicators of ecosystem health, yet current monitoring techniques cannot keep pace with the scale required to understand the impacts of deforestation, climate change, and habitat loss.

Our Solution:

Develop an automated bird sound classification system using machine learning that can:

- Process massive audio datasets
- Operate in real-world, noisy environments
- Provide real-time, remote biodiversity insights

Impact:

A scalable, AI-driven system like this can revolutionize how we:

- Track endangered species
- Detect ecological changes early
- Support conservation globally—even in inaccessible areas like rainforests or high mountains

Existing SOLUTIONS

- Automated Recording Units (ARUs) are used for remote data collection in ecological studies.
- Xeno-Canto & Macaulay Library – Large open databases of bird sounds used for research and training models.
- Merlin Bird ID (Cornell Lab) – Uses pre-trained ML models to identify bird species from audio recordings.
- BirdNET – AI-powered bird sound recognition system designed for citizen science and conservation.
- Bioacoustic Monitoring Systems – Hardware + software solutions for real-time bird call detection in the wild.
- Deep Learning Models (CNNs, RNNs, Transformers) – Applied for automated species classification with varied success in noisy conditions.

the GAP

- Limited Accuracy in Noisy Environments – Struggles with background noise and overlapping calls.
- Difficulty in Identifying Rare Species – Lack of sufficient training data for endangered/extinct birds.
- Limited Real-Time Detection – Most models work offline and are not optimized for real-time monitoring.
- Regional Bias in Datasets – Existing models perform better in well-documented regions but struggle in underrepresented areas.
- Lack of User-Friendly Citizen Science Tools – Current tools require expertise, limiting accessibility for non-experts.

Our solution

- Noise-Resistant Model – Employs advanced filtering techniques to handle background noise. (Bandpass filtering)
- Real-Time Detection – Enables instant bird species identification for field research and conservation.
- Augmented Dataset for Rare Species – Uses data augmentation and synthetic samples to improve accuracy.
- User-Friendly Interface – Develops a mobile/web tool for easy use by researchers & citizen scientists.

Literature survey

S No.	TITLE	YEAR	JOURNAL/CONFERENCE	LOCATION	METHODS	ACCURACY	AUTHOR	CITATION
1.	BirdNET: A Deep Learning Solution for Bird Sound Recognition	2018	Ecological Informatics	Elsevier Journal	CNN + RNN on spectrograms	81%	Kahl, S., Wood, C.M., Eibl, M., Klinck, H.	Kahl et al., 2021
2.	MERLIN Bird ID: AI-Powered Bird Identification	2019	Cornell Lab of Ornithology	Ithaca, New York, USA	Deep learning with audio-visual inputs	85%	Cornell Lab Team	Cornell Lab of Ornithology, 2019
3.	Warblr: Crowdsourced Bird Song Recognition	2016	Bioacoustics Journal	Taylor & Francis, UK	Machine learning (Random Forest, SVM)	78%	Stowell, D., Wood, M., Pamuła, H., Stylianou, Y., Glotin, H.	Stowell et al., 2016
4.	Xeno-Canto: Open-Source Bird Call Dataset	2015	Community Database	Global / Online	Crowdsourced data + traditional classifiers	NA	Xeno-Canto contributors	Xeno-Canto Foundation, 2023
5.	Automated Bird Sound Recognition using Attention-Based Neural Networks	2021	IEEE Transactions on Audio	Interspeech 2021, Brno, Czech Republic	Transformer-based models	88%	Gong, Y., Chung, Y.-A., Glass, J.	Gong et al., 2021

Passive Acoustic Monitoring for Avian Species

This paper provides a comprehensive review of how passive acoustic monitoring (PAM) techniques have been used for bird species identification.
Traditional methods involve manual spectrogram analysis and expert labeling, which is time-consuming and prone to human error.
Machine learning, especially deep learning models like CNNs and RNNs, has improved classification accuracy.

Challenges: Difficulty in distinguishing multiple overlapping calls, high noise levels in field recordings, and the need for extensive labeled data.

How this relates to our work: We are tackling the same challenges—distinguishing multiple bird calls, handling noise, and working with limited labeled data. Our approach will build on deep learning techniques while integrating new strategies like noise suppression and multi-label classification.

Species Detection Framework Using Automated Recording Units

Automated Recording Units (ARUs) are used for remote data collection in ecological studies.
This paper discusses a detection framework specifically for the critically endangered Jerdon's Courser.
The authors used spectrogram analysis and feature extraction to improve classification accuracy.
Challenges: ARUs capture a lot of environmental noise, and there is a need for robust preprocessing methods to filter irrelevant sounds.
Industry Application: ARUs are widely used in biodiversity monitoring programs to collect long-term data on species presence.
How this relates to our work:

Our dataset consists of ARU recordings, so we must address similar noise-related challenges.
Instead of manual feature extraction, we will use deep learning models that automatically learn features from spectrograms.
The concept of multi-species detection is crucial for our project, and we will refine it further using multi-label classification models.

Machine Learning for Soundscape Analysis

This paper reviews various ML techniques applied to ecological soundscapes, including CNNs, RNNs, and spectrogram-based classifiers.
Key Algorithms Used: Convolutional Neural Networks (CNNs) for feature extraction, Long Short-Term Memory (LSTM) networks for sequence modeling, and attention-based models for improving classification accuracy.
Challenges:
Class imbalance due to rare species having fewer samples.
Differences in background noise across different recording locations.
Performance limitations of models in real-world environments.
Industry Application: The use of AI-driven acoustic monitoring has been adopted by organizations working on species conservation, particularly for tracking biodiversity loss due to climate change.

How this relates to our work:
We plan to use CNN-based spectrogram classifiers but also explore attention-based models to enhance classification performance.
The issue of class imbalance will be tackled using techniques like data augmentation and class-weighted loss functions.
We aim to improve existing methodologies by adding our Song Richness Index (SRI), which quantifies species diversity in an area.

Deep Learning for Bird Sound Classification

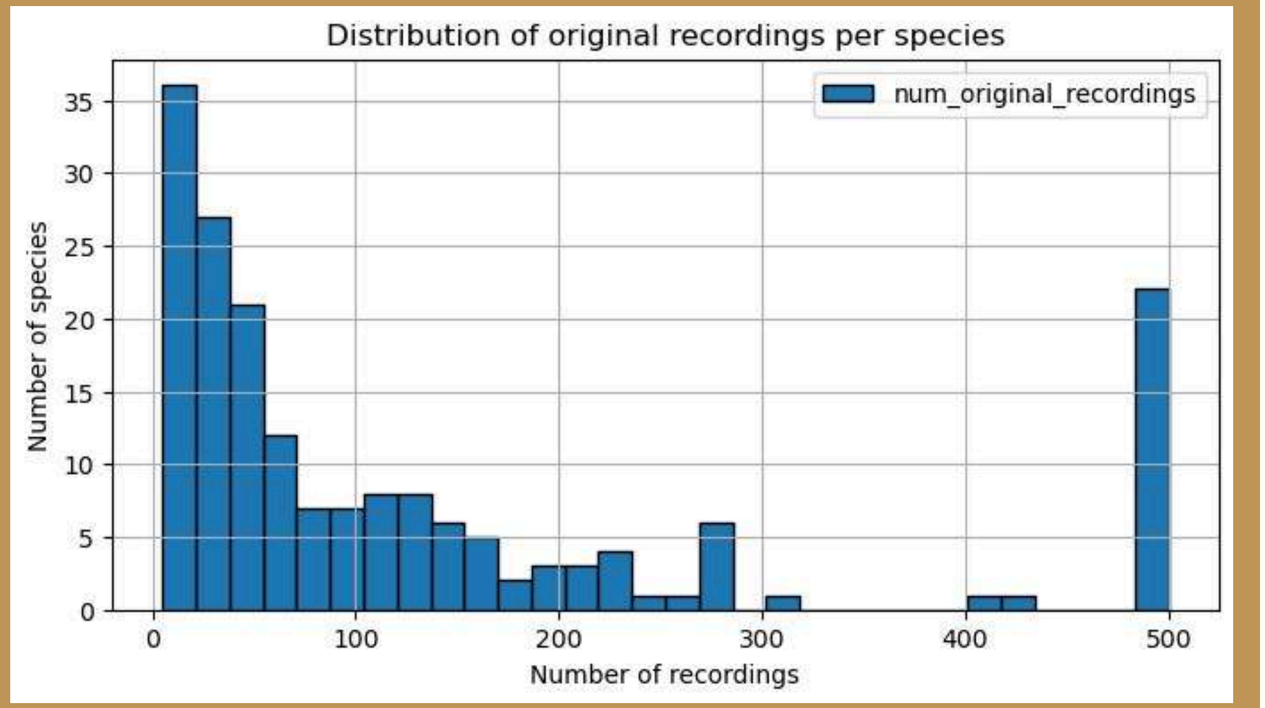
This study explores the application of deep learning models for bird call classification.
Pre-trained models such as VGGish and ResNet were used for feature extraction.
Transfer learning improved classification accuracy, especially for species with limited training data.
Challenges:
Training deep learning models requires extensive computational resources.
Annotated datasets are often limited, leading to overfitting.
Industry Application: AI-powered bird call recognition is being integrated into conservation tools and mobile apps for citizen science projects.
How this relates to our work:

We will use transfer learning to fine-tune pre-trained models on our dataset.
Since we also face the challenge of limited training data for rare species, we will experiment with few-shot learning techniques.
The paper highlights model optimization challenges, which we will address by fine-tuning architectures like EfficientNet.

DATASET

```
Checking dataset in: E:/birdclef-2024
Total species in 'train_audio': 182
Total files in 'test_soundscapes': 1
Total files in 'unlabeled_soundscapes': 8444
```

```
🔹 XC319059.wav: Sample Rate = 32000 Hz, Duration = 30.07 sec
🔹 XC840882.wav: Sample Rate = 32000 Hz, Duration = 42.82 sec
🔹 XC663615.wav: Sample Rate = 32000 Hz, Duration = 75.48 sec
🔹 XC692989.ogg: Sample Rate = 32000 Hz, Duration = 30.01 sec
🔹 XC338206.wav: Sample Rate = 32000 Hz, Duration = 61.05 sec
```

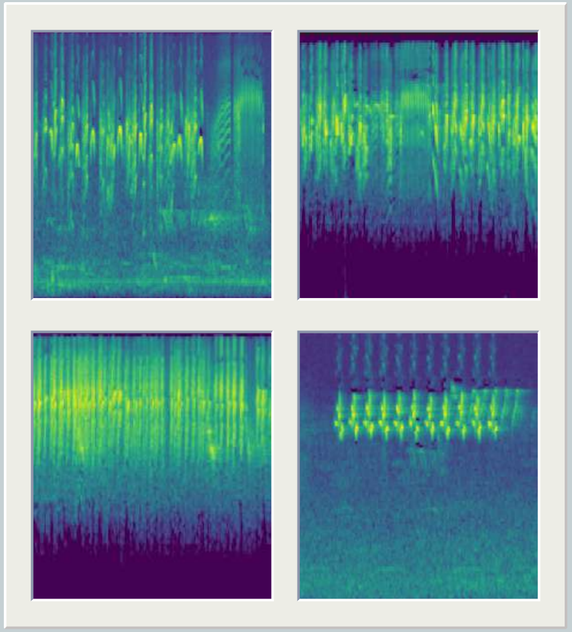


- We used BirdCLEF Dataset
- Contains 182 types of bird species
- Has 45000 instances across 182 indistinct classes
- train_audio: The training data consists of short recordings of individual bird calls generously uploaded by users of xenocanto.org.
- unlabeled_soundscapes: Unlabeled audio data from the same recording locations as the test soundscapes.
- train_metadata.csv: A wide range of metadata is provided for the training data.
- sample_submission.csv: A valid sample submission

DATA

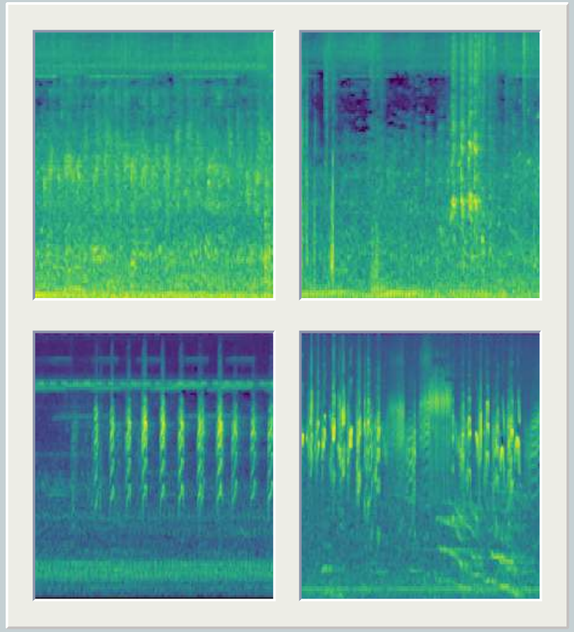
A Closer Look

Each bird species has a unique frequency range, so we can train the model to recognize different frequency patterns.



SPECTROGRAM

- Barn Swallow
  Hirundo rustica
- A fairly large, colorful swallow.
- Listen for dry, scratchy “svit svit” calls.



SPECTROGRAM

- Mindanao Lorikeet
- An uncommon, medium-sized, rather long-tailed parrot
- Voice consists of shrill, high-pitched shrieks.

WHY WE CHOSE THIS DATASET

Dataset Overview (Kaggle – Indian Bird Sound Classification)

- Source: Kaggle competition dataset (Western Ghats bird soundscape)
- Why we chose it:
  - Focused on endemic and endangered Indian bird species
  - Captures real-world audio challenges: overlapping calls, class imbalance, noise
  - Supports conservation goals in a biodiversity hotspot
- How data was collected:
  - Passive Acoustic Monitoring (PAM) — non-invasive, 24/7 audio recording
  - Organized by: IISER Tirupati, Cornell Lab, Google Research, LifeCLEF, Chemnitz University, Xeno-Canto
- Ethical considerations:
  - No human/animal interference — non-invasive data collection
  - Audio is anonymized with no sensitive metadata
- Dataset structure:
  - Thousands of audio clips, segmented into 5–10 second chunks
  - Each clip labeled with species ID (where available)
- Features & datapoints:
  - Main input: Mel-spectrograms (128 frequency bins × 512 time steps)
  - Tens of thousands of datapoints
  - Enhanced using SpecAugment, pitch/time shifting, and noise mixing

DATA PREPROCESSING RESULTS

1.Converted audio files from .ogg to .wav for compatibility and consistency.

2. Standardized all audio files to 32kHz sampling rate to ensure uniform processing.

3. Split long recordings into 5-second non-overlapping chunks for manageable model input.

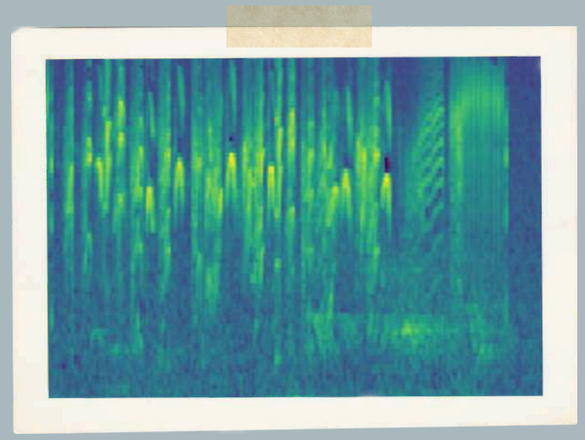
4. Transformed each 5-second audio chunk into a Mel spectrogram image for CNN-based models.

5. Identified and flagged nighttime audio chunks using audio features or a classifier.

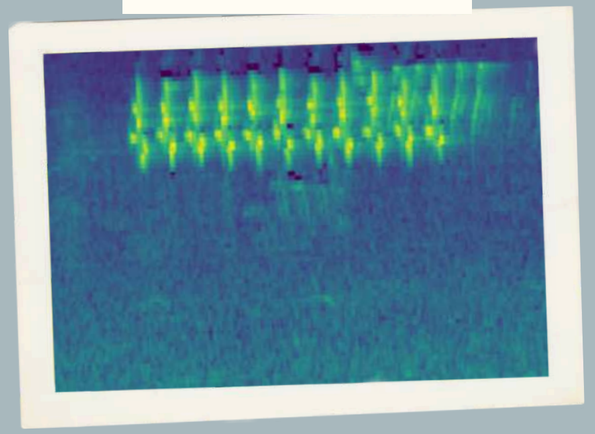
6.Filtering by Species Category
Created specific training datasets:
General balanced set (~40k samples across 182 species)
Rare species-only set (58 classes)
Nocturnal species set (7 classes)

7. Created cleaned and unified CSVs (train_split.csv, train_split_mixed.csv) linking each spectrogram to its label(s) and filepath.

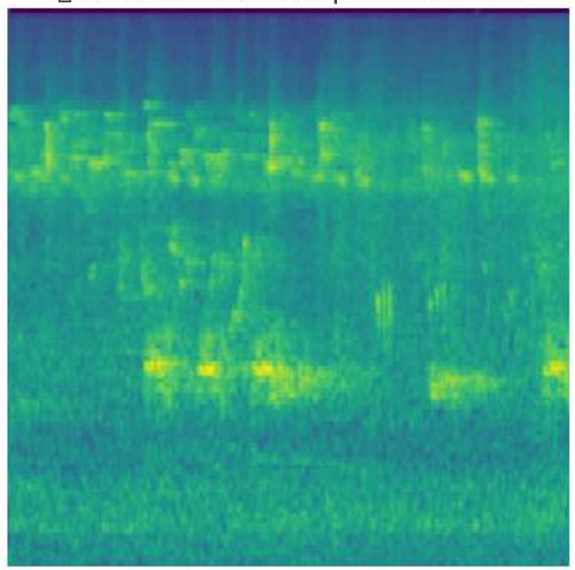
8. Unlabeled Soundscape Handling
Extracted up to 10 non-silent 5s segments per soundscape file.
Generated Mel spectrograms from these for inference.



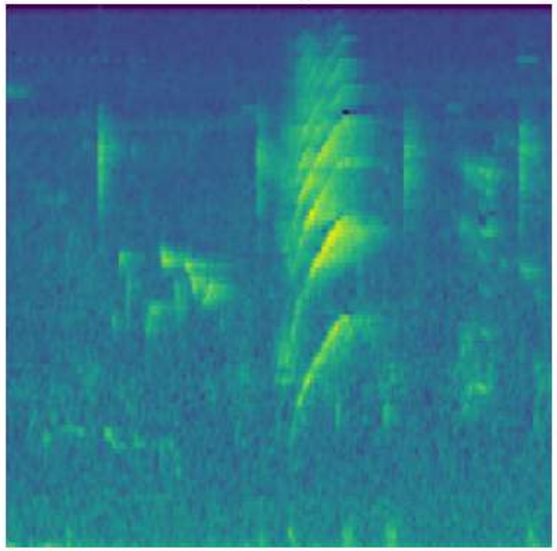
SPECTROGRAM



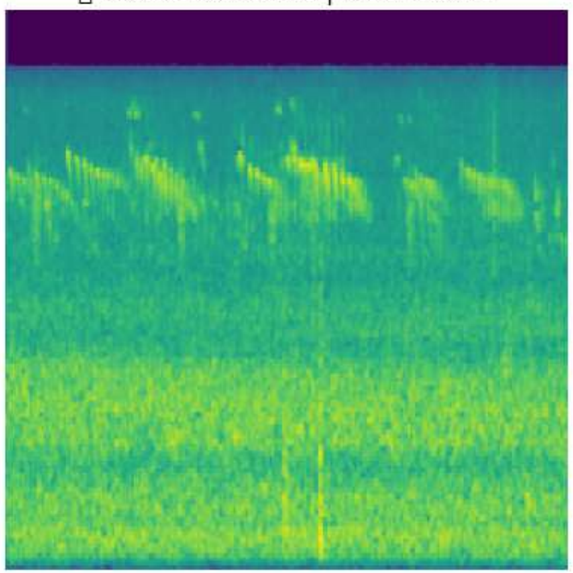
SPECTROGRAM



False Classification



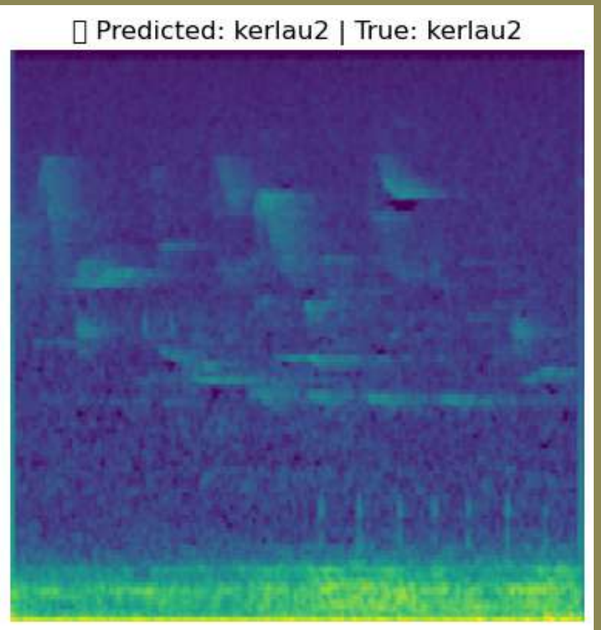
False Classification



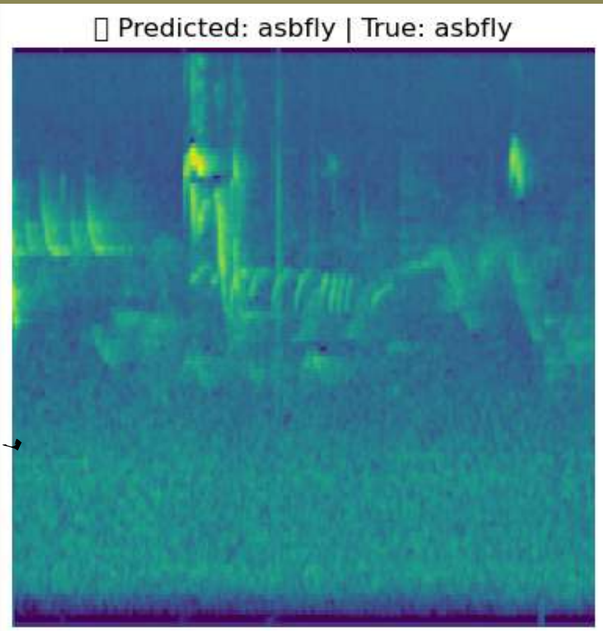
True Classification

OUTPUT

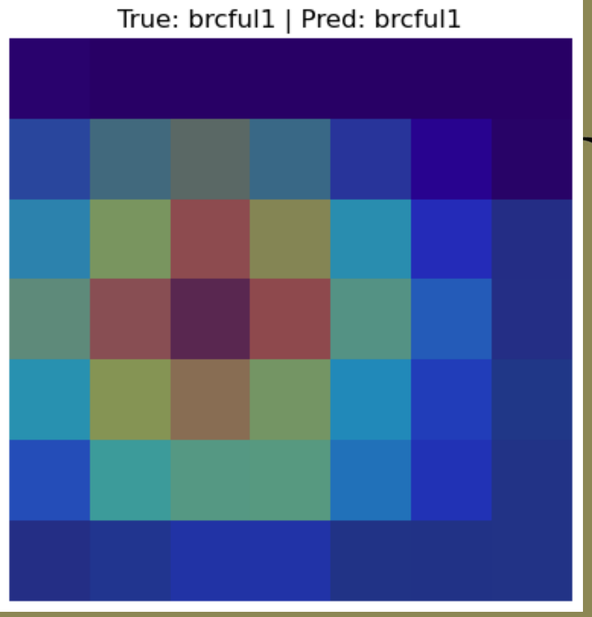
True Classification



True Classification

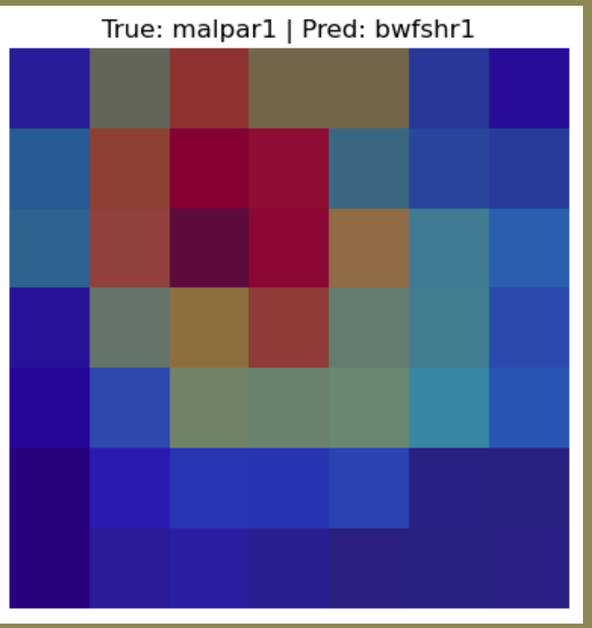


Grad-Cam



True Classification

False Classification



Why Grad-CAM?

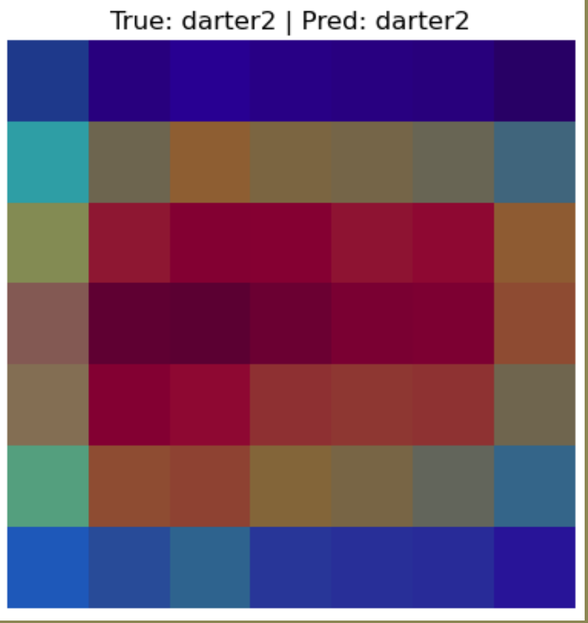
- Grad-CAM (Gradient-weighted Class Activation Mapping) helps visualize which parts of the spectrogram influenced the model's prediction.
- Useful for model interpretability and debugging misclassifications.

What This Shows:

- The heatmaps highlight regions in the spectrogram that the model focused on to make its decision.
- Warmer colors (reds) indicate higher attention by the model.

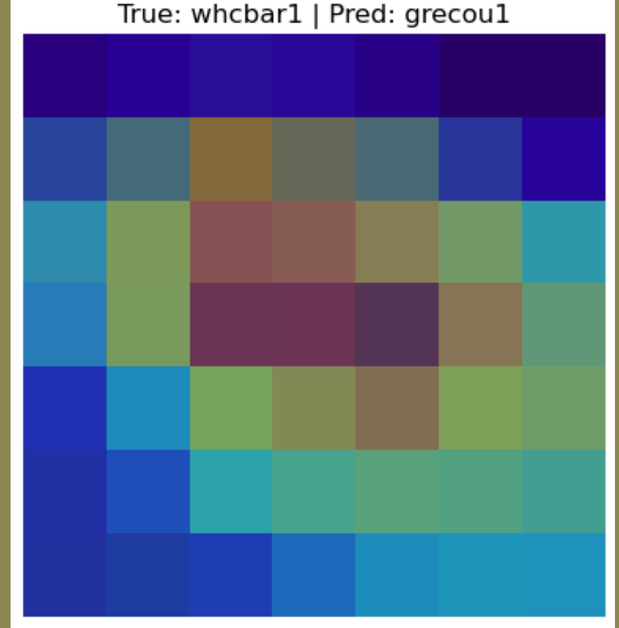
Insights:

- Helps identify if the model is learning relevant audio patterns.
- Misclassifications often show diffuse or misplaced attention, suggesting data similarity or label confusion.
- Confirms that the model is learning from tonal and temporal features rather than noise.



True Classification

False Classification



Proposed methodology

PLANNING BEHIND BIRDSHAZAM!!

1. CNNs for Bird Call Classification
(Used deep CNNs like ResNet and EfficientNet for image-based recognition of bird spectrograms.)

2. Multi-model Approach
(Trained separate models for general, nocturnal, and rare bird species for improved accuracy.)

1.Sound Preprocessing
(Applied silence trimming, bandpass filters, and log-scaled normalization)

2.Transfer Learning
(Fine-tuned ImageNet-pretrained models to handle low-data and rare-class situations)

1.Optimized Training
(Used weighted BCE loss, threshold tuning, early stopping, and oversampling for rare classes.)

2.Noise-Aware Segmentation
(Extracted clean 5s chunks from long recordings based on non-silent thresholds)

CHALLENGES FOR BIRDSHAZAM

Challenge	Why it's a Problem	How we plan to solve it
Limited labelled data for rare species	Most endangered species have very few recordings	Use few shot learning and self supervised learning techniques
Noisy nocturnal recordings	Background noise (wind, insects,traffic) reduces accuracy	Train multi label classifiers and Sound Event Detection models
Multi-species overlapping calls	Standard classifiers assume only one species per file	Use pretrained models & Sound Event Detection models
High computational cost	Transformers require large datasets & expensive GPUs	Use pretrained models & distillation for lightweight deployment
Real World Deployment	Most ML models are not optimized for mobile/web applications	Convert models using TensorFlow Life or PyTorch Mobile for real time use

Performance Metrics

BIRDSHAZAM RESULTS!!!!

Efficientnet_b0

1. Accuracy: **52.8%** (The proportion of correctly predicted labels over all predictions)

2.Precision: **55%** (The proportion of correctly detected among all detections)

3. Recall: **53%** (The proportion of correctly detected among all actual items in the dataset)

4. F1-Score: **0.52**

Resnet-50

1. Accuracy: **64%** (The proportion of correctly predicted labels over all predictions)

2.Precision: **66%** (The proportion of correctly detected among all detections)

3. Recall: **64%** (The proportion of correctly detected among all actual items in the dataset)

4. F1-Score: **0.64**

Custom_CNN

1. Accuracy: **49.15%** (The proportion of correctly predicted labels over all predictions)

2.Precision: **53%** (The proportion of correctly detected among all detections)

3. Recall: **50%** (The proportion of correctly detected among all actual items in the dataset)

4. F1-Score: **0.49**

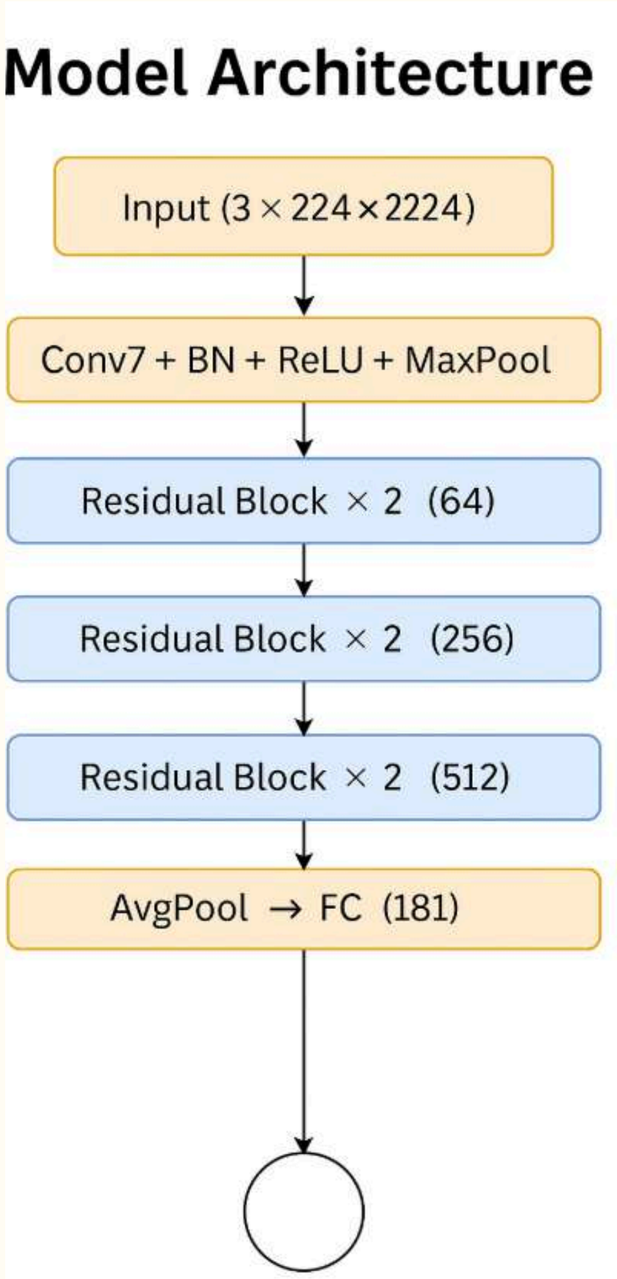
WHY A CUSTOM RESNET LIKE CNN?

Why a Custom CNN?

- Designed a lightweight ResNet-like architecture to learn features from mel spectrogram images of bird calls.
- Avoided transfer learning to fully control architecture depth, regularization, and parameter tuning.
- Useful for educational insights and benchmarking against pretrained models (EfficientNet, ResNet).

Architecture Details:

- Based on ResNet principles with skip (residual) connections.
- Composed of:
  - Initial Conv+BN+ReLU+MaxPool entry block
  - 4 residual stages (64→128→256→512 channels)
  - Adaptive Average Pooling → Fully Connected layer
- ResidualBlock includes:
  - Two 3×3 convolutions with batch normalization and ReLU
  - Optional downsampling using 1×1 convolution to match dimensions



EFFICIENTNET_B3

Why EfficientNet-B3?

- Chosen for its excellent accuracy vs. parameter efficiency tradeoff.
- Scales depth, width, and resolution in a compound manner.
- Performs well on spectrogram-based audio classification tasks with relatively fewer parameters.

- **Model:** EfficientNet-B3 (from timm library)
- **Loss Function**: Cross Entropy Loss
- **Optimizer:** Adam (learning_rate=1e-4)
- **Learning Rate Scheduler:** ReduceLROnPlateau (adaptive reduction on stagnating loss)
- **Batch Size**: 32
- **Epochs**: Up to 6 with early stopping

Training Enhancements

- Used PyTorch + TQDM for real-time progress tracking
- Validation Accuracy & Training Loss logged after each epoch
- Model checkpointing:
  - Model saved after every epoch
  - Best model saved as best_efficientnet_b3_model.pth

Memory & Resource Management

- Manual cleanup of tensors after each batch:
  - del images, labels, outputs
  - torch.cuda.empty_cache()
  - gc.collect() for memory release

Performance Metrics

ACCURACY 82%	Overall correct predictions on validation set
PRECISION 0.81	Correct positive predictions out of f all predicted positives
RECALL 0.81	Correct positive predictions out ouf all actual positives
F1-SCORE 0.81	Harmonic mean of precision and recall

NIGHT_TIME_DETECTION

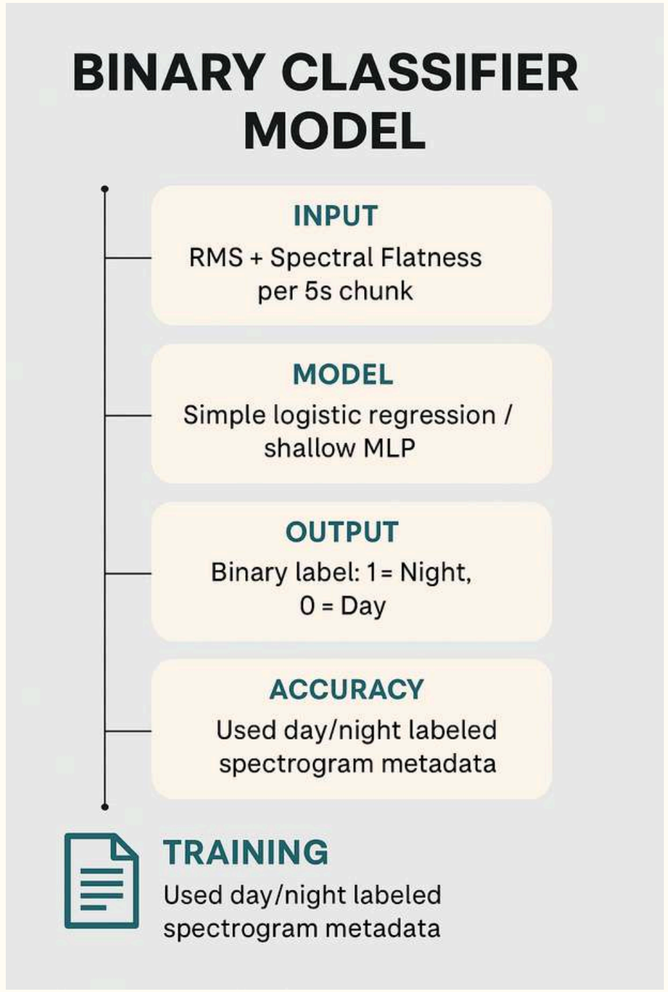
Why Nighttime Detection?

- Some bird species (e.g., owls) only vocalize at night.
- Running the nocturnal model on irrelevant daytime chunks introduces noise and reduces precision.
- Solution: Train a binary classifier to flag whether a spectrogram comes from a nighttime segment or not.

Features Used for Detection

1. RMS Energy (Root Mean Square)
   - Measures the intensity of the audio signal.
   - Nighttime segments often have low RMS due to less environmental noise.

1. Spectral Flatness
   - Measures tonal vs noisy nature of the sound.
   - Flatness near 1 → white noise; Flatness near 0 → tonal sound.
   - Night segments often have higher flatness due to background noise or silence.



	filename	flatness	rms	night_predicted
649	bfly\XC589132_part5.wav	0.028339453041553497	0.01104509737342596	0
650	bfly\XC589132_part6.wav	0.029808923602104187	0.01189398393034935	0
651	bfly\XC596043_part1.wav	0.17279092967510223	0.005670992191880941	1
652	fly\XC596043_part10.wav	0.20161588490009308	0.000553359161131084	1
653	fly\XC596043_part11.wav	0.17300918698310852	0.002508403034880757 3	1
654	fly\XC596043_part12.wav	0.16736671328544617	0.0019082255894318223	1
655	fly\XC596043_part13.wav	0.1594221293926239	0.001152980257757008	1
656	fly\XC596043_part14.wav	0.1454283744096756	0.002122869249433279	0
657	fly\XC596043_part15.wav	0.16091440618038177	0.0019088290864601731	1
658	fly\XC596043_part16.wav	0.1732221394777298	0.0006619623163715005	1
659	fly\XC596043_part17.wav	0.17181181907653809	0.0006509292288683355	1
660	fly\XC596043_part18.wav	0.1670485883951187	0.000666784355416894	1
661	fly\XC596043_part19.wav	0.15944500267505646	0.0006766448495909572	1
662	bfly\XC596043_part2.wav	0.16329480707645416	0.005188973154872656	1
663	fly\XC596043_part20.wav	0.1423746794462204	0.002726650331169367	0
664	fly\XC596043_part21.wav	0.14417678117752075	0.002323541324585676	0
665	fly\XC596043_part22.wav	0.15698359906673431	0.0025130445137619972	1
666	fly\XC596043_part23.wav	0.1897130161523819	0.0005841703386977315	1
667	fly\XC596043_part24.wav	0.183625265955925	0.0006061487947590649	1
668	fly\XC596043_part25.wav	0.16607734560966492	0.002292474964633584	1
669	fly\XC596043_part26.wav	0.1639997363090515	0.0030941744334995747	1
670	fly\XC596043_part27.wav	0.17189589142799377	0.0017567627364769578	1
671	fly\XC596043_part28.wav	0.1747671514749527	0.0027341537643224	1
672	bfly\XC596043_part3.wav	0.14254437386989594	0.005342910531908274	0
673	bfly\XC596043_part4.wav	0.1715199202299118	0.0029298162553459406	1
674	bfly\XC596043_part5.wav	0.2021990865468979	0.0005665207863785326	1
675	bfly\XC596043_part6.wav	0.14152711629867554	0.0008813220774754882	0
676	bfly\XC596043_part7.wav	0.13486753404140472	0.002380989259108901	0
677	bfly\XC596043_part8.wav	0.1648949533700943	0.0025921582709997892	1

NOCTURNAL_SPECIES

Why a Separate Model?

- Nocturnal birds vocalize primarily at night and are often drowned out by ambient night noise.
- Standard bird models trained on full-day data underperform on night recordings.
- Solution: Train a specialized classifier for nocturnal bird species using only night-segmented data.

Night Soundscape Relevance:

- Enhances classification of rare nocturnal birds often missed by general models
- Supports biodiversity research during low-light and night-time monitoring
- Integrated into inference pipeline using night_flagged_audio.csv to:
  - Automatically route nighttime chunks to this model
  - Merge predictions with general/rare species models using max-probability logic

Dataset Info:

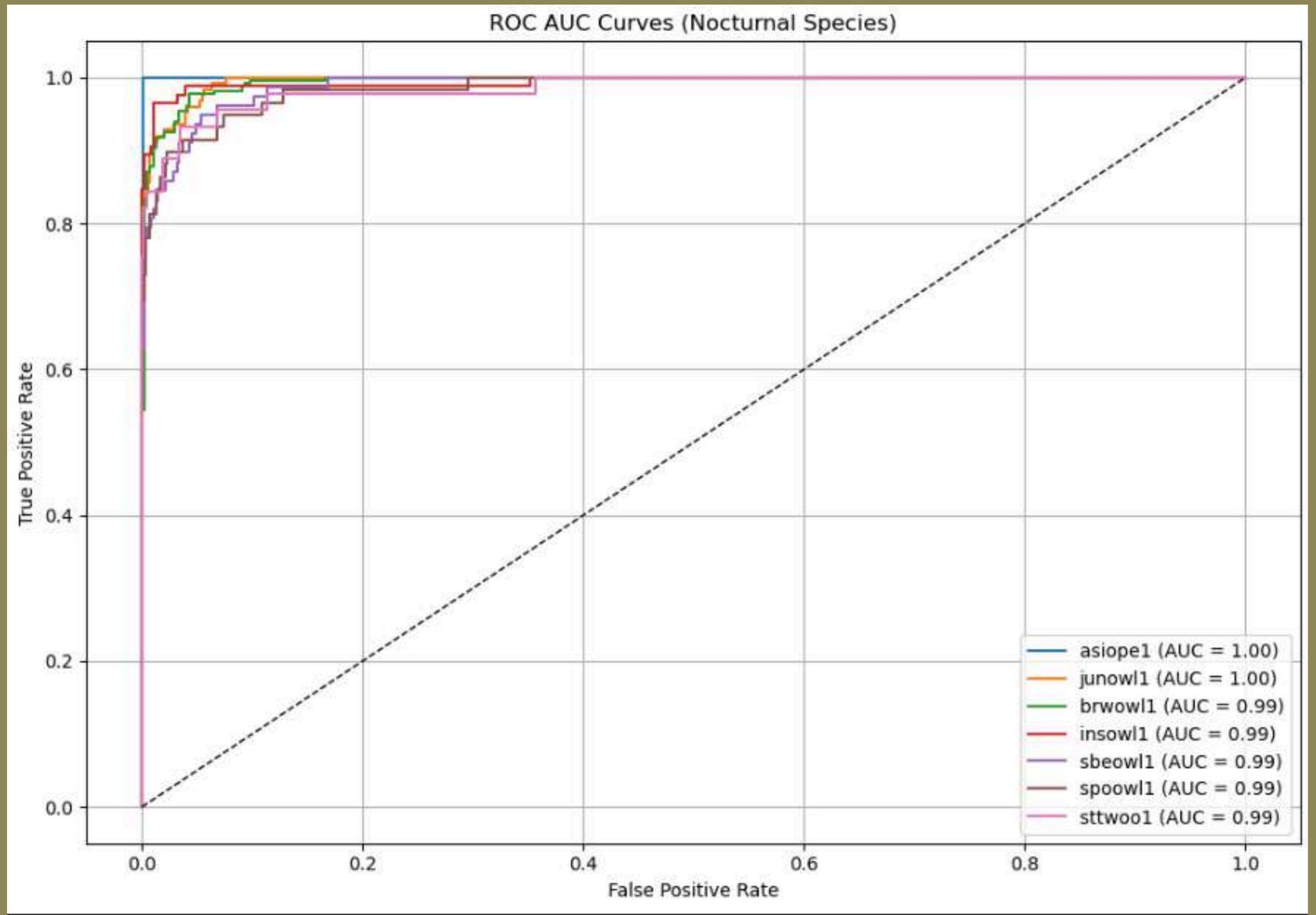
- **Total Samples**: 3,389
- **Number of Classes**: 7 nocturnal bird species
- **Species:**
  - asiope1 (Indian Eagle Owl); brwowl1 (Brown Wood Owl) ;insowl1 (Indian Scops Owl)
  - junowl1 (Jungle Owlet); spoowl1 (Spot-bellied Owl) ;sbeowl1 (Short-eared Owl)
  - sttwoo1 (Spotted Owlet)

- **Model:** EfficientNet-B3
- Used cross-entropy loss, Adam optimizer, and learning rate scheduler
- **Learning Rate Scheduler:** ReduceLROnPlateau (adaptive reduction on stagnating loss)
- **Batch Size**: 32
- **Epochs**: Up to 10 with early stopping

Evaluation Metrics

Nocturnal Species Model (EfficientNet-B3)

Validation Accuracy Overall correct predictions on validation set	91.74 %
Precision Correct positive predictions out of all predicted positives	0.92
Recall Correct positive predictions out of all actual positives	0.83
F1-Score Harmonic mean of precision and recall	0.86
No. of Classes	7
Total Samples	3,389
Best Model	best_nocturnal_model.pth




```
Macro-Average ROC AUC: 0.9921
Micro-Average ROC AUC: 0.9941
```

RARE_SPECIES

Why a Separate Rare Species Model?

- Many rare or endangered bird species have very limited training data in the full dataset.
- General classifier tends to overfit dominant species, ignoring underrepresented ones.
- Solution: Train a specialized classifier focused only on 58 rare species to ensure better sensitivity and performance.

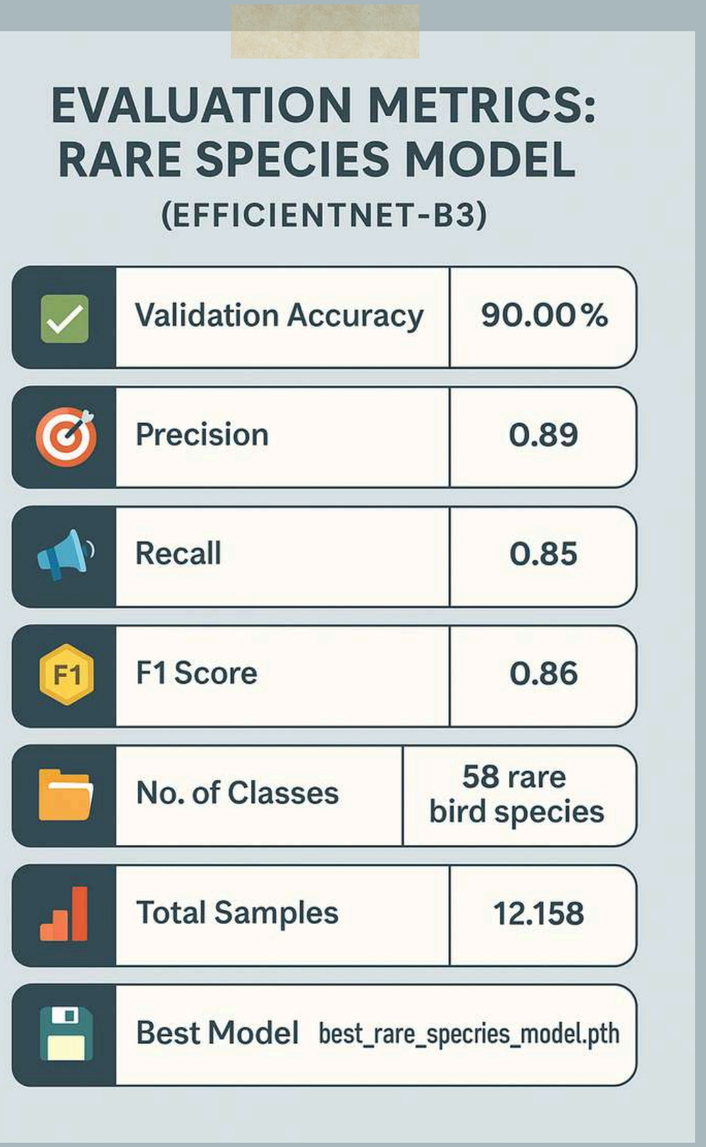
Dataset Details:

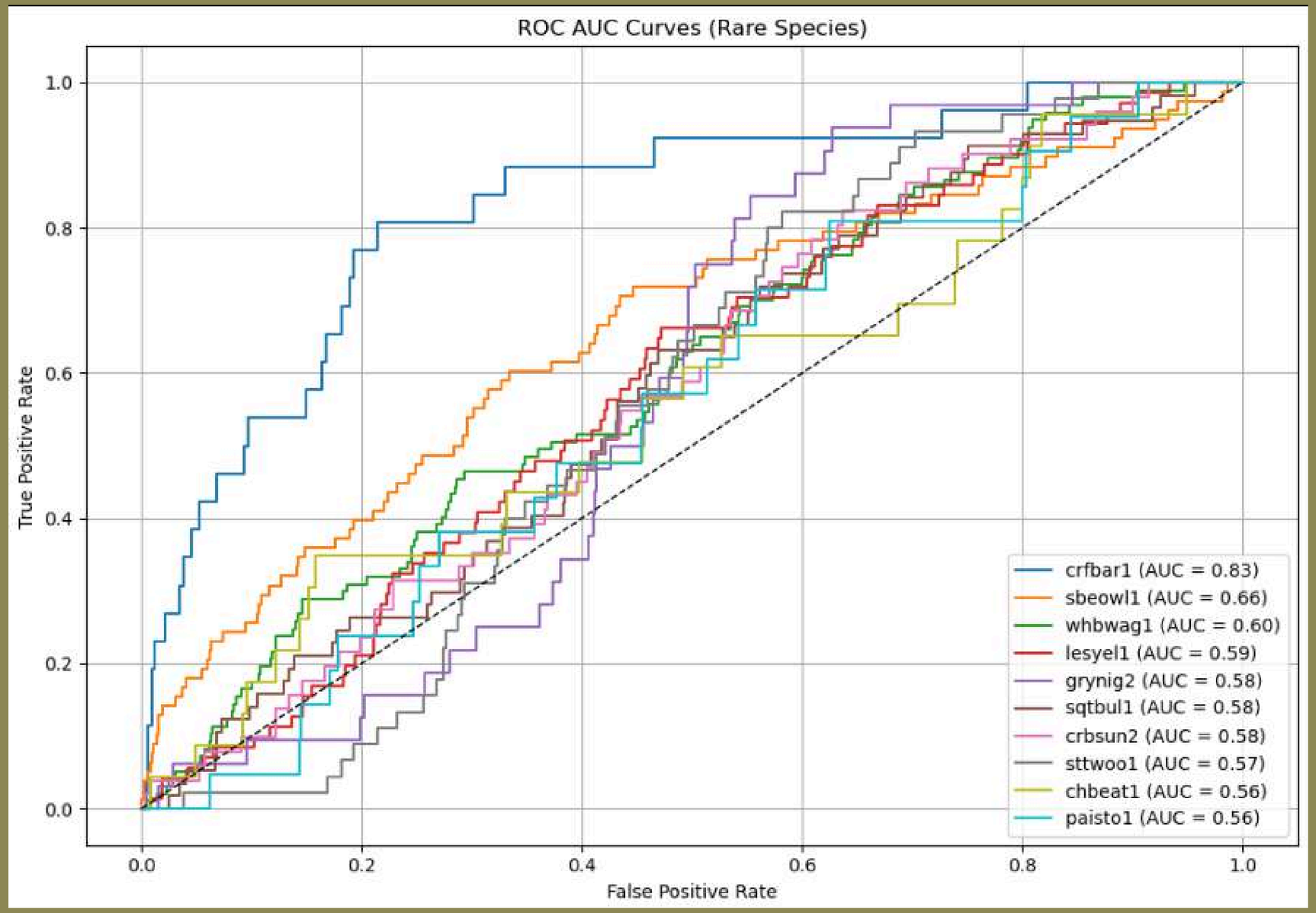
- **Total Samples:** 12,158
- **Classes:** 58 rare or endangered bird species
- **Data Selection**: Filtered and oversampled only those species with lowest sample frequency

How It Fits Into the System:

- Used during inference along with general and nocturnal models
- Predictions for rare classes merged into the final probability vector
- Uses probability max logic per class (i.e., highest from any model is retained)

- **Model:** EfficientNet-B3
- Used cross-entropy loss, Adam optimizer, and learning rate scheduler
- **Learning Rate Scheduler:** ReduceLROnPlateau (adaptive reduction on stagnating loss)
- **Batch Size**: 32
- **Epochs**: Up to 10 with early stopping




ROC AUC Curves (Rare Species)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
crfbar1 (AUC = 0.83)
sbeowl1 (AUC = 0.66)
whbwag1 (AUC = 0.60)
lesyel1 (AUC = 0.59)
grynig2 (AUC = 0.58)
sqtbul1 (AUC = 0.58)
crbsun2 (AUC = 0.58)
sttwoo1 (AUC = 0.57)
chbeat1 (AUC = 0.56)
paisto1 (AUC = 0.56)

MULTI_LABEL

Why Multi-Label?

- Real-world bird soundscapes often contain multiple species calling simultaneously.
- Traditional single-label classifiers fail to capture overlapping species.
- This model allows multiple species to be predicted for a single 5s audio segment (mel spectrogram).

Dataset Details:

- Total Samples: 15,000
- Labels per sample: 1 to 3 bird species per spectrogram
- Input: 300×300 Mel Spectrogram images
- Output: Binary vector of length 181 (1 if species present, else 0)

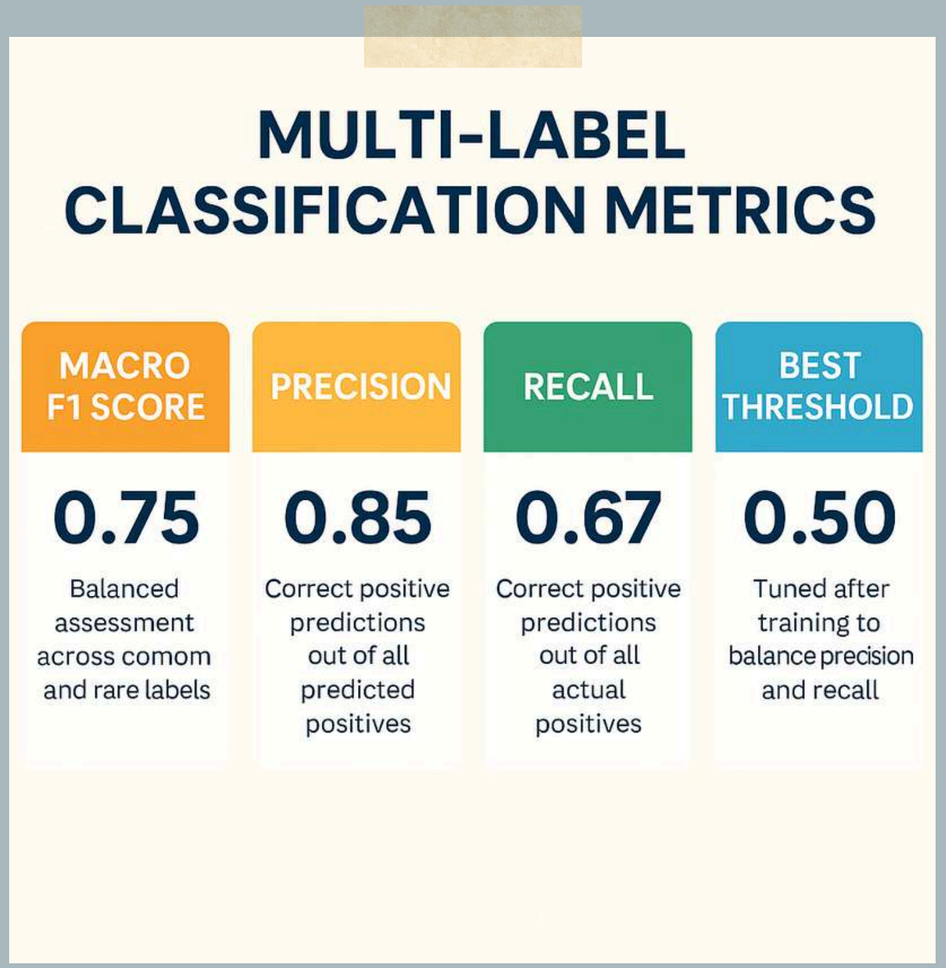
Label Generation:

- Synthetic mixing using spectrogram overlay (mixup) technique
- Combined single-label samples into multi-label training images
- Targets represented as binary vectors instead of single class indices

How It Fits Into the System:

- Model is used in parallel with single-label models during inference
- Supports real-time detection of multiple co-occurring species
- Final output combines:
- General model predictions |Rare & nocturnal model outputs| Multi-label predictions

- **Model:** EfficientNet-B3
- **Loss Function**: BCEWithLogitsLoss (multi-label binary loss)
- **Label Smoothing:** Applied (0.05) to reduce overfitting
- **Optimizer:** Adam
- Training Epochs: 10



UNLABELED SOUNDSCAPE

Goal:

- Predict bird species present in real-world field recordings with no labeled data.
- Use trained models to estimate bird presence probabilities across 81,000 audio segments.

Input Dataset:

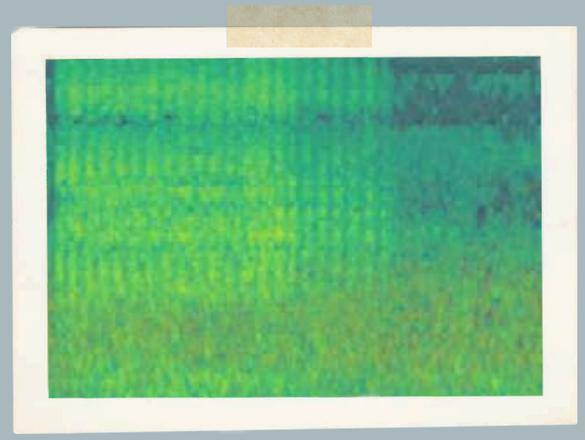
- 81,000 audio samples extracted from long soundscape recordings
- Each sample: 5-second chunk → Mel Spectrogram (300×300)
- Nighttime vs Daytime tagged using a binary classifier (night_flagged_audio_unlabeled.csv)

Preprocessing:

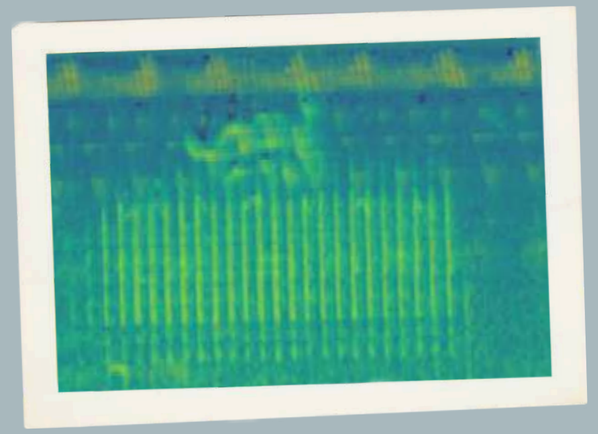
- Same preprocessing as training:
  - .ogg → .wav
  - Resample to 32 kHz
  - Segment into 5s clips
  - Generate Mel Spectrograms

Output Format:

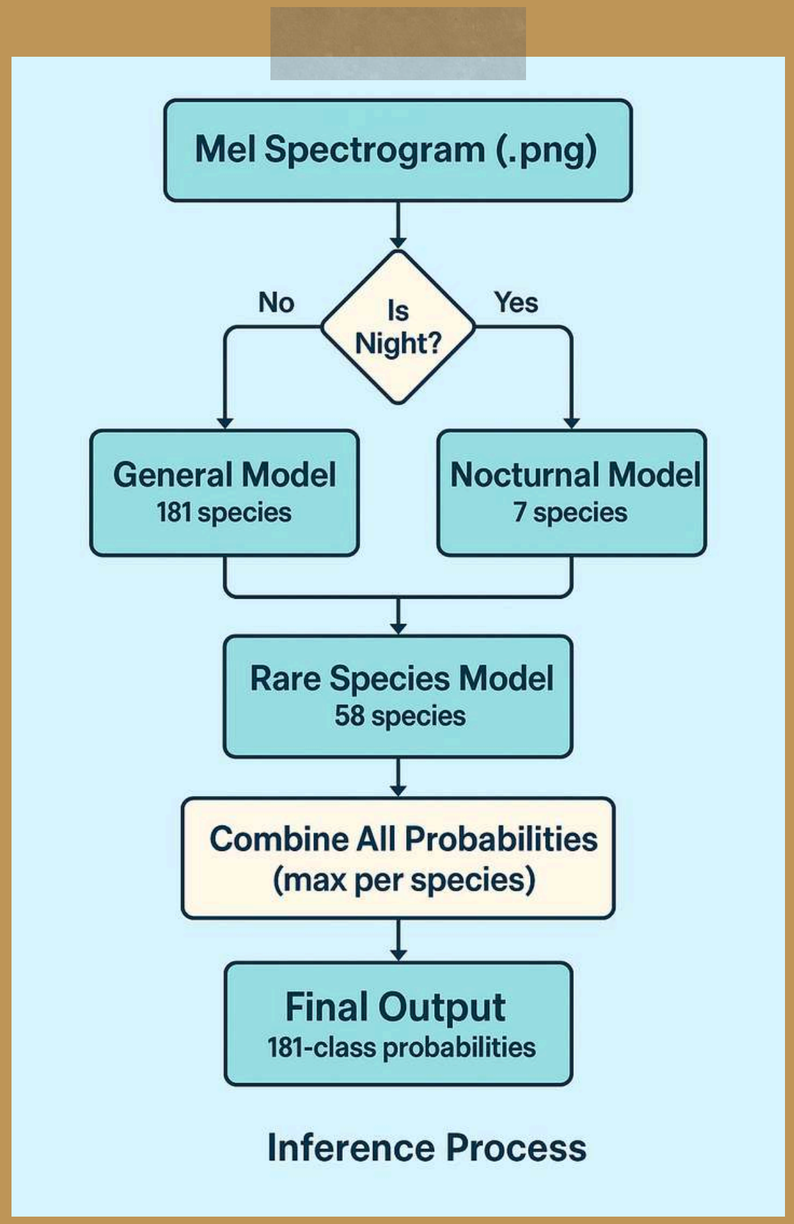
- Rows: 81,000 samples
- Columns: Probability for each of the 181 bird species
- Values represent the likelihood of a species being present in the 5-second chunk



SPECTROGRAM



SPECTROGRAM



Models Used:

- General Model (181 species)
- Nocturnal Model (7 species; only on night-flagged samples)
- Rare Species Model (58 rare/extinct bird species)

Routing Logic:

1. ***Nighttime Detection***
   - Custom model flags segments as nighttime or daytime
   - Based on Spectral Flatness, RMS etc.
   - Inference then routed accordingly
2. ***Model Inference***
   - General Model: Always applied (181 species)
   - Nocturnal Model: Applied only to night-tagged chunks (7 species)
   - Rare Species Model: Applied to all chunks (58 species)
3. ***Probability Fusion Logic***
   - For each of 181 bird classes:
     - If predicted by multiple models, use the maximum of their probabilities
   - Ensures no loss of confidence for rare/nocturnal species
4. ***CSV Output Structure***
   - Columns: filename, followed by class_0 to class_180
   - Each cell: Probability (0–1) of that species being present in that chunk

general_probs[idx_] = max(general_probs[idx_], rare/nocturnal_prob)

	row_id	soundscape_id	t_min	asbfly	ashdro1	ashpri1	ashwoo2
1	1000170626_0_5	1000170626	0	0.00013875254	0.0002520115	3.6085745e-05	4.874162e-05
2	1000170626_10_15	1000170626	10	0.00039332345	0.0005491374	0.0001650275	0.00028338443
3	1000170626_15_20	1000170626	15	0.00044145368	0.00011450552	0.00020093366	0.00018564328
4	1000170626_20_25	1000170626	20	6.861043e-05	2.5084795e-05	2.26767e-06	1.8331928e-05
5	1000170626_25_30	1000170626	25	0.00084874395	0.0035366896	0.000657504	0.001343052
6	1000170626_30_35	1000170626	30	0.00052005216	3.6421836e-05	5.161438e-05	0.00017216834
7	1000170626_35_40	1000170626	35	0.0004096827	1.2570265e-05	1.837484e-05	2.3731596e-05
8	1000170626_40_45	1000170626	40	7.938179e-05	6.149346e-06	2.393793e-05	7.748455e-06
9	1000170626_45_50	1000170626	45	0.0017378402	8.06628e-05	1.1093383e-05	1.050641e-05
10	1000170626_5_10	1000170626	5	0.0010004613	0.0010146461	4.570103e-05	0.00027961654
11	1000308629_0_5	1000308629	0	0.00083766965	0.00409413	0.0005407409	0.00055006205
12	1000308629_10_15	1000308629	10	0.0032325985	0.0059492844	0.0017058261	0.00011123237
13	1000308629_15_20	1000308629	15	0.00078884675	0.012597843	0.0022527801	0.0013972832
14	1000308629_20_25	1000308629	20	0.0006359572	0.0009801134	0.0007028531	0.001486241
15	1000308629_25_30	1000308629	25	0.0003505857	0.0031405718	8.181915e-05	0.01799271
16	1000308629_30_35	1000308629	30	0.00043464385	0.00029748882	0.0048710685	5.5725846e-05
17	1000308629_35_40	1000308629	35	0.00019632888	0.0001305011	0.0012780854	0.00026303358
18	1000308629_40_45	1000308629	40	0.00026964836	0.00010519466	0.0015296439	1.2657408e-05
19	1000308629_45_50	1000308629	45	0.00015104559	0.0010992999	0.007824803	0.00010384905
20	1000308629_5_10	1000308629	5	0.013422782	0.0017765113	0.00036236178	0.00014363519
21	1000389428_0_5	1000389428	0	0.0009112453	0.00022575229	0.00017029345	2.4664023e-05
22	1000389428_10_15	1000389428	10	0.0009696232	0.00019050064	0.0001733911	0.00039803924
23	1000389428_15_20	1000389428	15	0.0012899993	9.1519505e-05	0.00065776694	0.00019287782
24	1000389428_20_25	1000389428	20	0.011285192	8.0875456e-05	0.00028859577	0.00022232959
25	1000389428_25_30	1000389428	25	0.00040671098	1.1649008e-06	0.0009815036	3.0565975e-06
26	1000389428_30_35	1000389428	30	0.00011174085	3.9240913e-06	8.523778e-05	8.76714e-06
27	1000389428_35_40	1000389428	35	0.00029075268	3.198432e-05	0.00035199622	8.805214e-06

- After running inference on 81,000 unlabeled audio samples, we generated a prediction CSV where:
- Each row = a 5-second audio spectrogram chunk
- Each column = one of the 181 bird species
- Each cell = probability that the species is present in that chunk (range: 0 to 1)
- An official result CSV containing the true presence probabilities of each species per file (privately evaluated),
- This allows us to cross-validate our model’s performance by comparing our predicted CSV with theirs (if accessed via competition backend).

Real-World Impact

- **Wildlife Conservation**

→ Monitor endangered and migratory bird species in protected ecosystems like the Western Ghats, Himalayas, and national parks.

- **Urban Biodiversity Tracking**

→ Deploy in cities and university campuses to observe changes in avian activity amidst urban development.

- **Citizen Science Engagement**

→ Encourage public participation by allowing users to upload bird audio via mobile apps and contribute to crowdsourced labeling.

- **Ecological Research & Policy**

→ Use Song Richness Index and species presence data to support habitat planning, conservation policy, and biodiversity reports.

- **Education & Awareness**

→ Install interactive dashboards in schools, museums, and wildlife parks to educate users about local birdlife in real-time.

Challenges in Real-World Scaling

- **Data Volume Growth**
→ Continuous environmental recording generates massive audio data — requires cloud storage, on-device compression, or data pruning strategies.

- **Real-Time Inference Speed**
→ EfficientNet models are computationally heavy for edge devices — need quantization, model distillation, or lighter alternatives for real-time performance.

- **Species Expansion**
→ New or migratory species require ongoing model retraining and dynamic label mapping to remain accurate and relevant.

- **Environmental Noise Variability**
→ Background sounds like wind, traffic, rain affect predictions — calls for adaptive noise filtering or noise-aware training pipelines.

- **Label Scarcity for Rare Birds**
→ Many rare species are underrepresented in training data — may lead to misclassification or complete omission.

Thank you