Bird Sound Classification: Weekly Progress

Focus: Smarter Spectrograms, More Birds, Better Models

Leonardo Mannini

May 28, 2025

Project Goal & Core Workflow

- **Goal:** Efficient bird sound classification for edge devices (AudioMoth).
- **Key Stages:**
    1. Data Acquisition, Preprocessing & Augmentation
    2. Lightweight Model Design & Iteration
    3. Systematic Training & Evaluation
    4. Edge Deployment Preparation (Quantization, ONNX, TFLite)

Model: `Improved_Phi_GRU_ATT` (1/2)

- **Origin:** Adapted from a suite of lightweight architectures explored for the Keyword Spotting (KWS) task.
- **Our Starting Point:** Selected for its balance of feature extraction and temporal modeling for bird sound classification.
- **Base Architecture:** Hybrid CNN-RNN with Attention.
- **Designed for Efficiency:** Lightweight for edge deployment.

- **Key Components:**
  - **Spectrogram Input:** Converts raw audio to a 2D representation.
    - *Current:* Linear Triangular Filterbank Spectrograms.
  - **CNN Backbone (`MatchboxNetSkip`):** Extracts local features from spectrograms using depthwise separable convolutions.
    - Configurable: `base_filters`, `num_layers`.
  - **GRU Layer (`nn.GRU`):** Models temporal dependencies in the extracted features.
    - Unidirectional for efficiency (processes sequence chronologically). `hidden_dim: 32`.

Model: `Improved_Phi_GRU_ATT` (2/2)

- **Key Components (cont.):**
  - **Projection Layer (`nn.Linear`):** Further processes GRU output.
  - **Attention Mechanism (`AttentionLayer`):** Focuses on relevant parts of the audio sequence for classification.
  - **Classification Head (`nn.Linear`):** Outputs probabilities for `num_classes`.
- **Parameters:** Approx. **37k** with current configuration. Meaning 148KB with float32 or 74KB in float16 or 37KB in int8.

Initial Training Steps & Data Pipeline Debugging

Key challenges: dataset splitting, class imbalance, "non-bird" representation.
Initial SR: 22050 Hz, Spectrogram: Mel

Attempt 1: Binary Classification

- **Run:** `2025-05-11_19-41-43`
- **Setup:** 2 classes (1 bird + non-bird), `load_pregenerated_no_birds: true` (few samples).
- **Results:** Test Acc: ~99%
- **Observation:** High accuracy

Attempt 2: Multi-Class (4 Classes - Problematic)

- **Runs:** `2025-05-12_06-32-34` & `2025-05-12_15-40-01`
- **Setup:** 4 classes.
- **Issue:** Dataset splitting still problematic (Train/Val/Test all 3446 samples).
- **Results:** Test Acc: ~64.86%
- **Observation:** Performance drop.

Attempt 3: Corrected Splitting (4 Classes - 10 Epochs)

- **Run:** `2025-05-12_16-48-06`
- **Setup: Dataset splitting logic implemented!** 4 classes.
    - Train=3205, Val=1314, Test=1314.
- **Epochs: 10**
- **Results:** Test Acc: ~82.65% (Best Val Acc: ~82.88%)
- **Observation:** Dramatic improvement! However, "non-bird" handling (dynamic balancing) was not yet in place. This led to the next phase of improvements.

32kHz & Linear Spectrograms

- **Goal:** Preserve more high-frequency details potentially lost in Mel compression, crucial for some bird calls.
- **Higher Sample Rate (32kHz):**
  - Complements linear spectrograms by providing a wider frequency range.
  - Potentially captures more discriminative information for bird species with high-pitched calls.

- **Linear Spectrogram Types Explored:**
   i. **`linear_stft` (Raw STFT):** Uniform frequency resolution. Higher dimensionality (`n_fft // 2 + 1` bins).
   ii. **`linear_triangular` (Linear Triangular Filters):**
      - STFT + linearly spaced triangular filters.
      - Reduces dimensionality (e.g., to `n_linear_filters: 64`) while maintaining linear emphasis.
      - **Current Best Choice:** Good balance of performance and model input size.

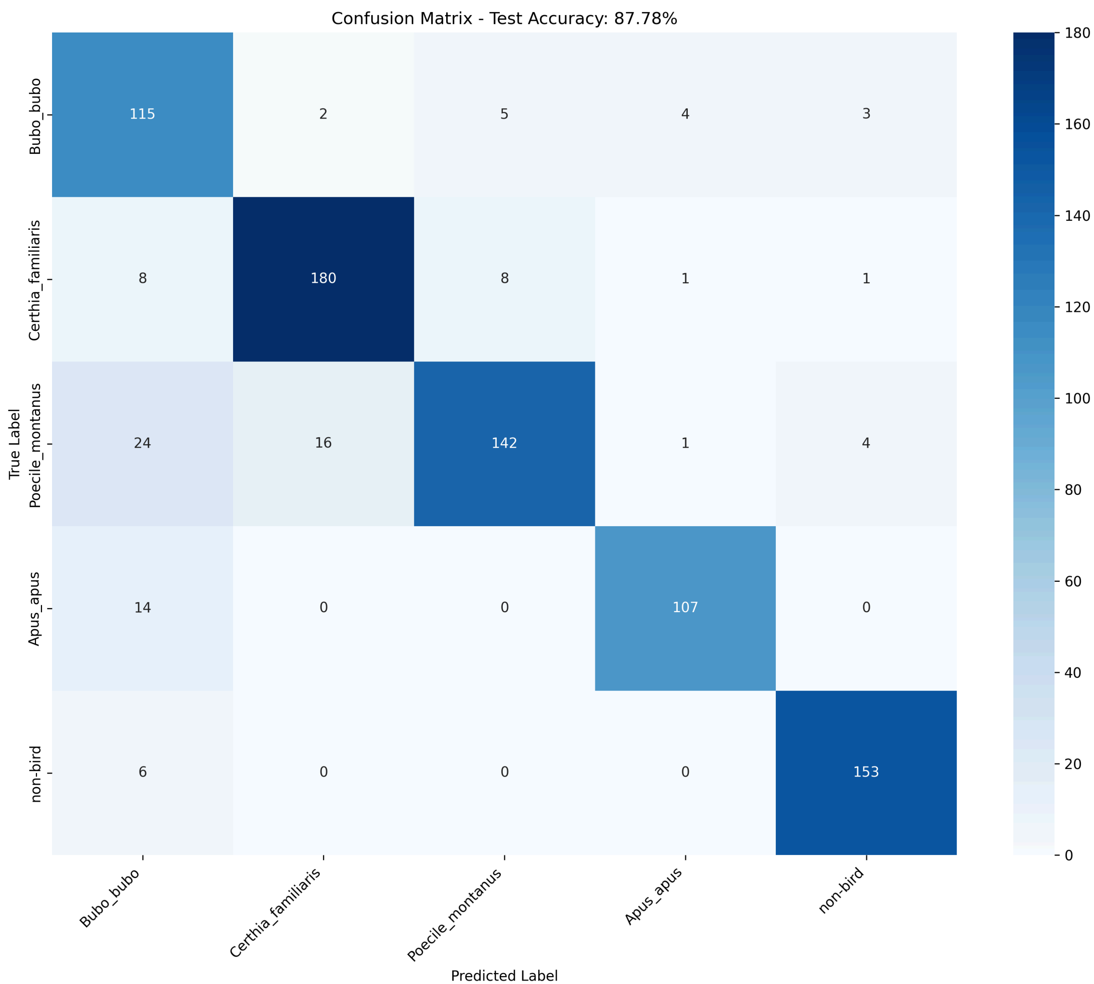
Training Evolution with 32kHz & Linear Triangular Specs

- **Focus:** 4 Bird Classes (`Bubo_bubo`, `Certhia_familiaris`, `Poecile_montanus`, `Apus_apus`) + `non-bird`.
- **Spectrogram:** `linear_triangular` (`n_linear_filters: 64`, `n_fft: 1024`, `hop_length: 320`, `sr: 32kHz`).
- **Dynamic balancing**: Also, a dynamic balancing of the classes has been implemented, configurable via dhe hydra config file.

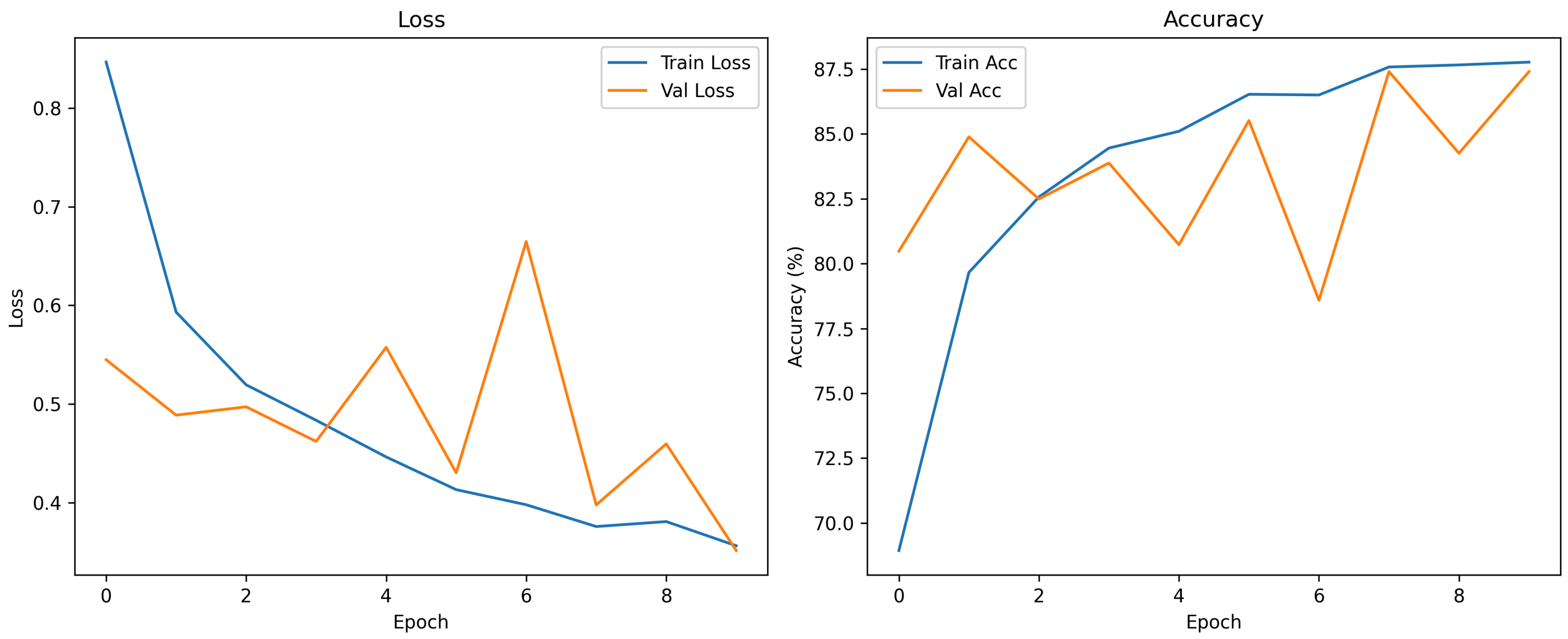
Run: `4birds_CF_linear_triangular_10epochs`

- **Date:** 2025-05-13_21-08-37
- **Test Accuracy: 87.78%**
- **Observations:**
    - Good "non-bird" recall (~96%).
    - `Poecile_montanus` challenging (~76% recall).
    - Balanced performance for other birds (88-91% recall).

`10epochs`: Confusion Matrix



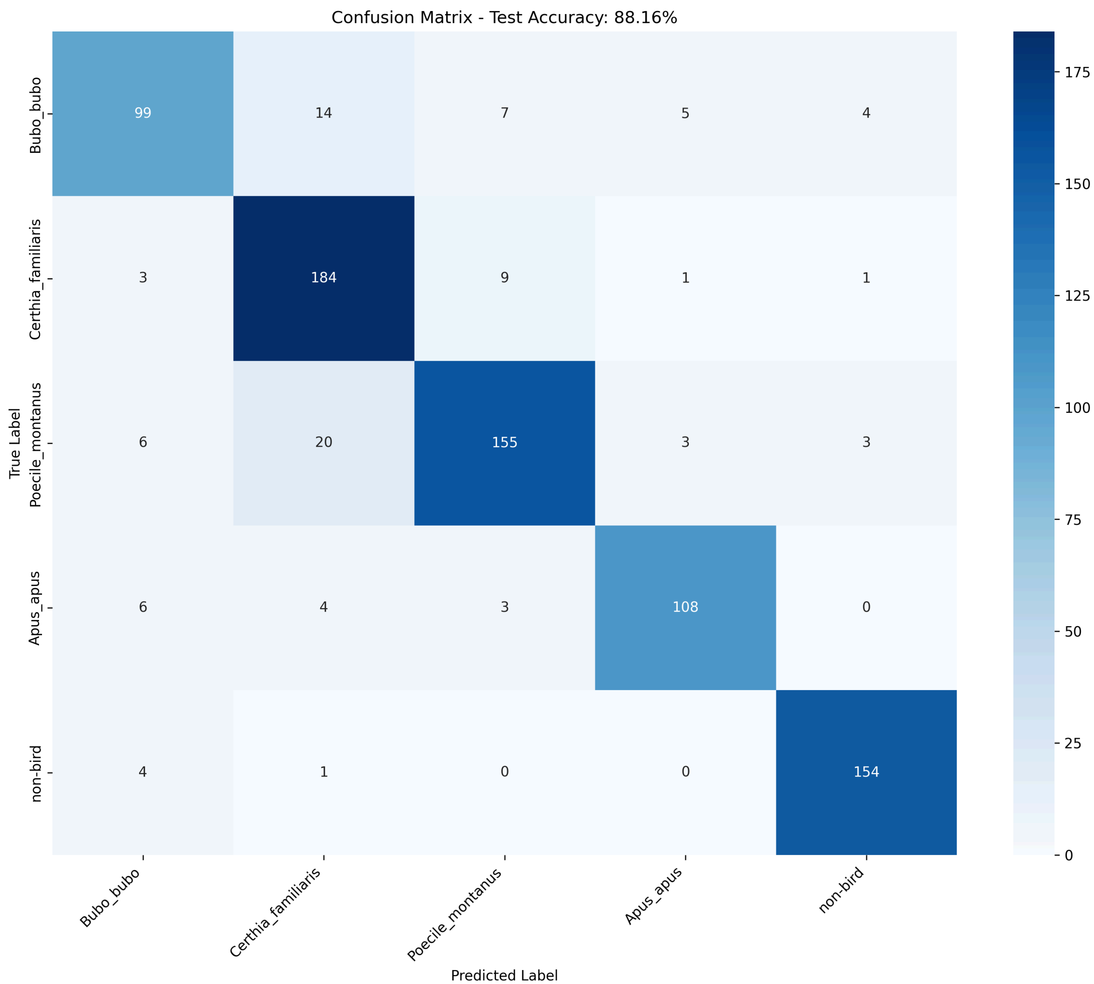
`10epochs`: Training History



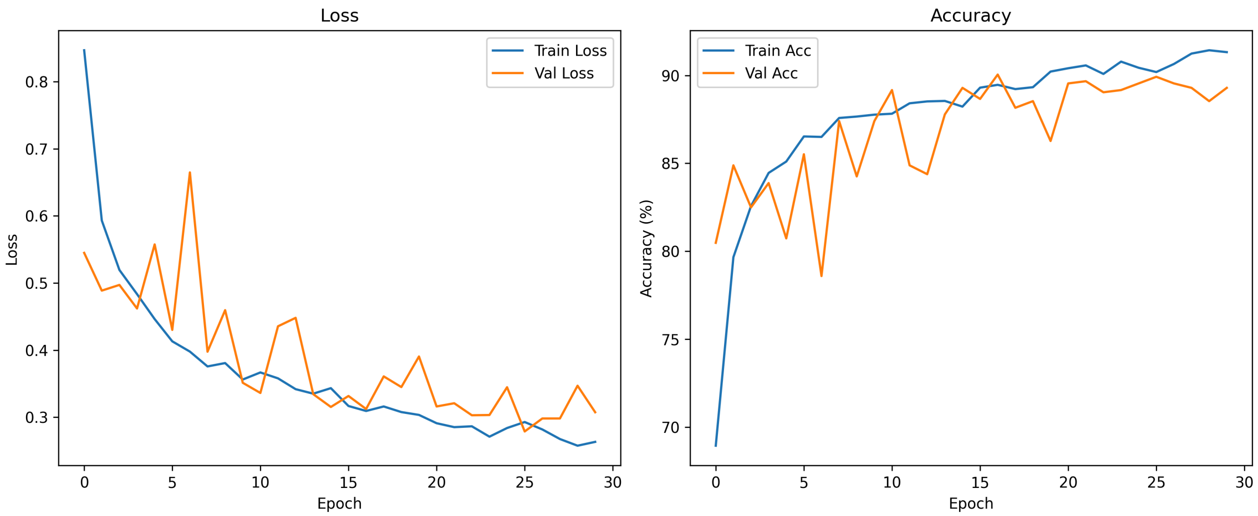
Run: `4birds_CF_linear_triangular_30epochs` (Early Stopping)

- **Date:** 2025-05-13_22-45-46
- **Best Val Acc: 90.05% (Epoch 17)**
- **Test Accuracy (Epoch 17 model): 88.16%**
- **Observations:**
    - Slightly better test accuracy.
    - Signs of overfitting after epoch 17.
    - `Bubo_bubo` recall decreased (77%), `Poecile_montanus` improved (83%).

`30epochs` (Best Val): Confusion Matrix



`30epochs`: Training History



Analysis of Recent Runs (10 vs 30 Epochs w/ Early Stop)

- **Overall Accuracy:** Slight gain with more epochs (87.78% -> 88.16%).
- **Problems:** Still many
- **Conclusion:** 10-epoch model currently appears more balanced overall. Further refinement needed for `Poecile_montanus` and `Bubo_bubo` stability.

Previous State (Recap from May 14th)

- **Model:** `Improved_Phi_GRU_ATT` (~37k params)
- **Task:** Recognition of 4 Bird Species + Non-Bird (5 Classes)
- **Spectrogram:** Fixed Linear Triangular Filterbank (32kHz SR)
- **Best Performance (4 Birds):** ~88% Test Accuracy

Progress Update: May 28th, 2025

- **Focus:**
    i. **Smarter Audio Features:** Making spectrograms adaptable.
    ii. **Harder Problem:** More classes (recognize more birds, focusing on the most "difficult" ones)
    iii. **Model Enhancements:** Improving model capacity and training.

Key Advancement 1: Adaptive Spectrograms

- **The Challenge:** Standard audio features (spectrograms) are fixed. Can the model learn to customize them for better performance?
- **Why it Matters:** Different bird sounds have unique frequency characteristics. This allows the model to tailor its "view" of the sound.
- **Impact:** The model actively adjusts these parameters, effectively learning a more optimal audio representation.

Different birds, different frequency patterns. Rather than setting fixed filters, we let the model learn **where** to focus.

Adaptive Spectrograms: Visualizing the Change

Before: Fixed Linear Filters

- Uniformly spaced triangular filters.
- Same shape across all frequencies.

Fixed Linear Triangular Filterbank

After: Adaptive Log-Linear Filters

- Model learns `breakpoint` & `transition_width`.
- Effectively creates different filter spacing/shapes for low vs. high frequencies.

Adaptive Combined Log-Linear Filterbank with Learned Parameters

Learnable parameters

The model now learns two key parameters to shape its audio input:

1. `breakpoint` **(Hz):** *Where* to switch from log-scale to linear-scale.
2. `transition_width` **(Hz):** *How* smoothly these two scales are blended.

Spectrogram Transformation: 1. Raw Audio Input

Raw STFT Spectrogram

Short-Time Fourier Transform (STFT) of the raw audio.

Spectrogram Transformation: 2. "Before" - Fixed Linear Filters

Linear Filtered Spectrogram

The raw STFT is processed by a standard, fixed linear filterbank. The frequency axis is now "Filter Index" – each horizontal band represents one filter's output.

Spectrogram Transformation: 3. "After" - Adaptive Log-Linear Filters

Adaptive Log-Linear Filtered Spectrogram

Here, STFT is processed by our adaptive log-linear filterbank. This version tends to emphasize the lower frequency bands, suggesting the model found more useful information there.

Why Adaptive Log-Linear? The Benefits

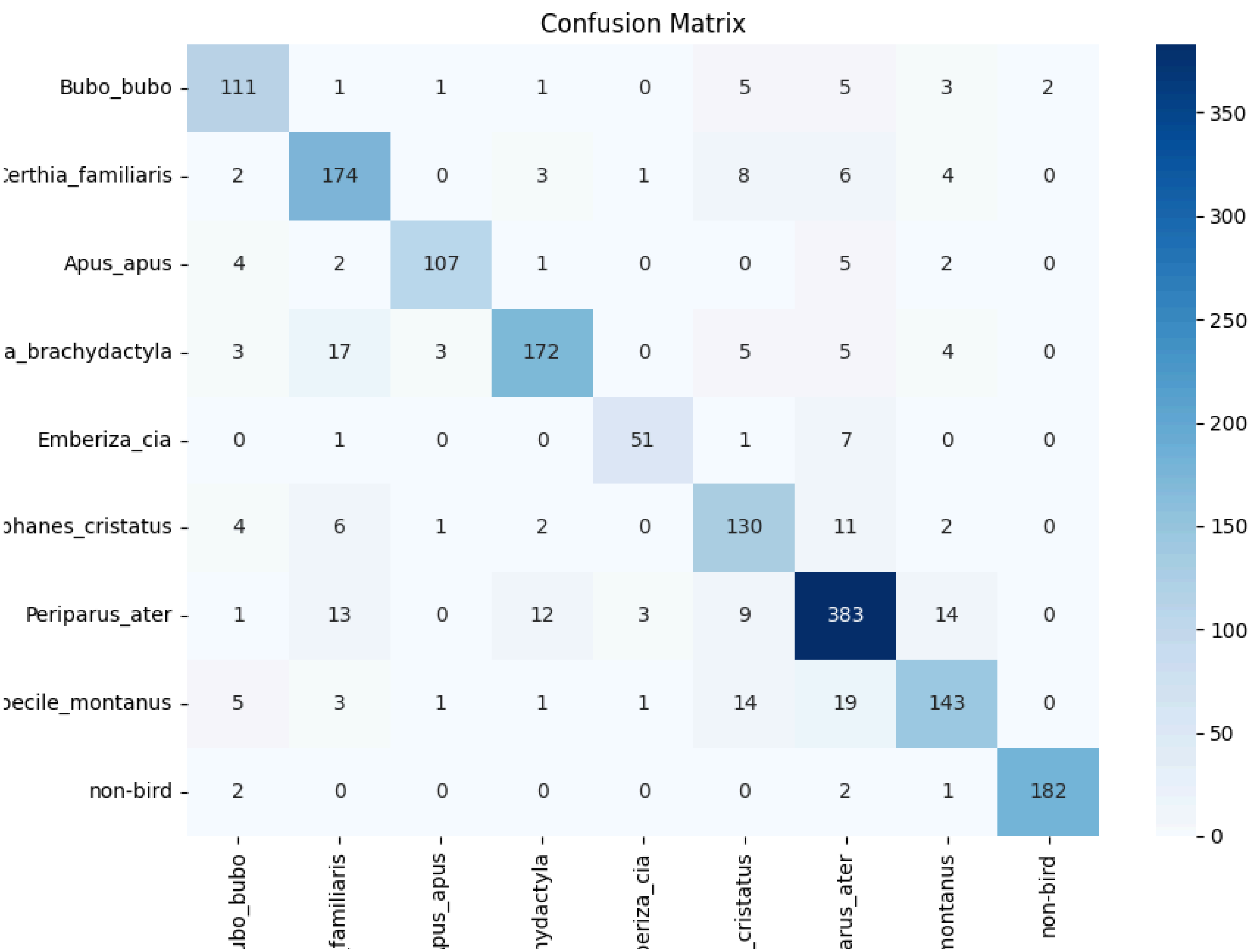
- **Focusing on What Matters (Low Frequencies):**
  - **Better Perceptual Resolution:** In lower frequency ranges, small absolute changes in Hz (pitch) are easily perceived as distinct notes.
  - This is crucial for harmonic and melodic sounds, like bird songs, where fundamental frequencies and early harmonics are key.
  - (Kinda like in *music*! 🦜 = 🎶)

Key Advancement 2: Scaling Up

- **Increased Model Capacity:**
  - Enhanced the GRU component (temporal modeling) by doubling its `hidden_dim` (32 -> **64**).
  - Bigger "memory" of the past (at the cost of bigger size and training time)
  - Model size grew modestly (~37k -> **~53.5k params**)
- **Expanded Task: More Birds!**
  - Moved from 4 bird species to **8 bird species** (+ non-bird), now a 9-class problem.
  - **New "difficult" Species Added**

Experiment: 50 Epochs (Faster Adaptation)

- **Results (Model from Epoch 45):**
  - **Test Accuracy: 86.39%** (same results as previous tests, but this time with more classes, focusing on "diffucult" ones)
  - Learned Breakpoint: Shifted further, from 4000 Hz to **~1905 Hz**.
  - Learned Transition Width: Adapted more, to **~65 Hz** (from 100 Hz).


Confusion Matrix
Bubo_bubo
:erthia_familiaris
Apus_apus
a_brachydactyla
Emberiza_cia
ohanes_cristatus
Periparus_ater
oecile_montanus
non-bird
111 1 1 1 0 5 5 3 2
2 174 0 3 1 8 6 4 0
4 2 107 1 0 0 5 2 0
3 17 3 172 0 5 5 4 0
0 1 0 0 51 1 7 0 0
4 6 1 2 0 130 11 2 0
1 13 0 12 3 9 383 14 0
5 3 1 1 1 14 19 143 0
2 0 0 0 0 0 2 1 182
ubo_bubo
familiaris
pus_apus
iydactyla
eriza_cia
_cristatus
arus_ater
nontanus
non-bird
350
300
250
200
150
100
50
0

Training History (50 Epochs)

Performance on 9 Classes (50 Epoch Run, Best Model)

(Recalls shown - How well does it identify each class?)

Species	Recall	Species	Recall
`Bubo_bubo`	86.0%	`Lophophanes_cristatus`	83.3%
`Certhia_familiaris`	87.9%	`Periparus_ater`	87.0%
`Apus_apus`	88.4%	**`Poecile_montanus`** (Still a bit Hard!)	**76.5%**
`Certhia_brachydactyla`	82.3%	`non-bird`	97.3%
`Emberiza_cia`	85.0%		

Overall Test Accuracy: 86.39%

Key Learnings & Current Status (May 28th)

- **Successes:**
  - **Adaptive spectrograms work!** Model learns to adjust audio features, improving performance (Test Acc: **86.39% on 9 classes**).
  - Model scales well to more classes and slightly larger size.

Next Steps: Roadmap Priorities

1. **Tackle `Poecile_montanus`:**
   - Deep dive: Try to understand why low recall
2. **Continue Scaling & Benchmarking:**
   - Train with all the bird species.
   - Compare against standard benchmarks (BirdNet).

Next Step: Knowledge Distillation

Leveraging an Expert Model to Teach Our Compact One

Knowledge Distillation

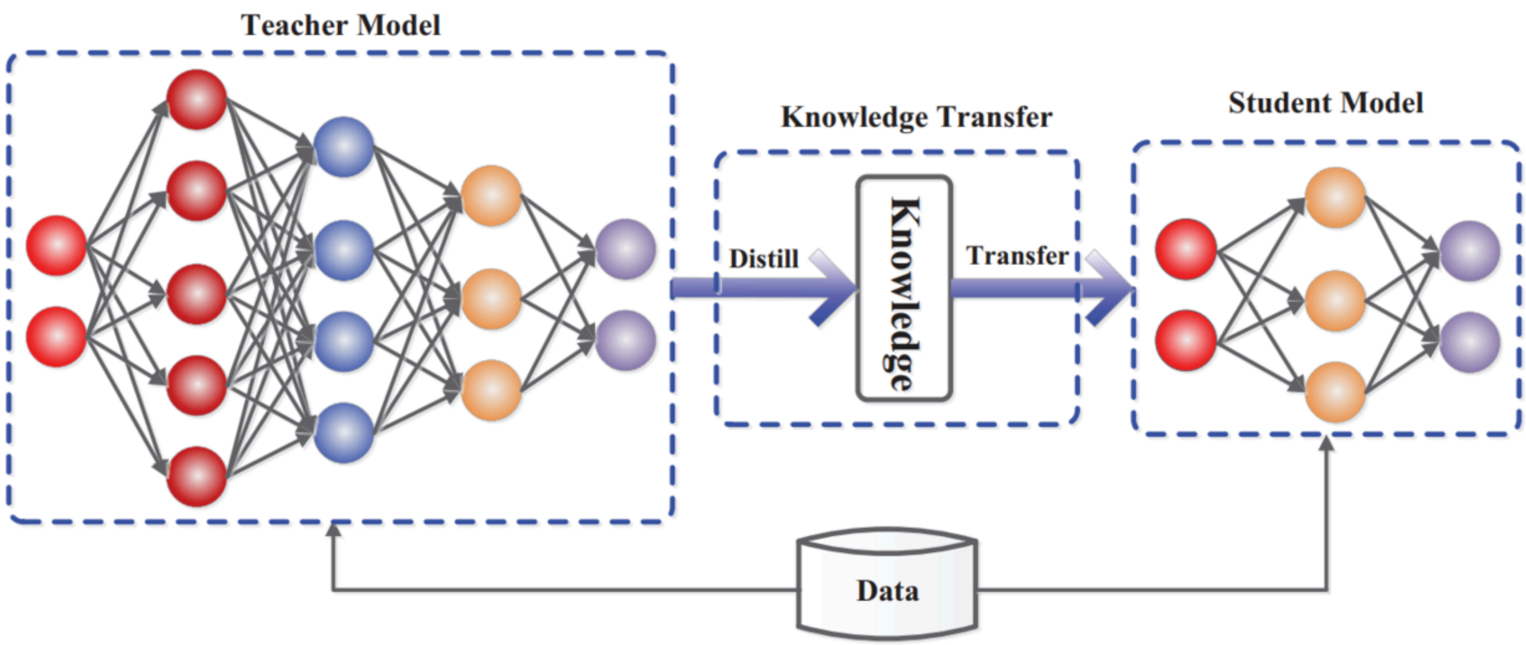
It's a "Teacher-Student" training strategy.

- **Teacher:** A large, high-performance model (e.g., BirdNet).
- **Student:** Our small, efficient edge model.

The student learns from two sources:

1. **Hard Labels:** The ground truth (e.g., "This is `Bubo_bubo`").
2. **Soft Labels:** The Teacher's detailed probability outputs (e.g., "I'm 70% sure it's `Bubo_bubo`, but it also sounds 20% like `Apus_apus`").

This transfers the "reasoning" of the teacher to the student.



Our Distillation Plan: Teacher vs. Student

Teacher: BirdNet

- State-of-the-art bird sound classifier.
- Recognizes over 3,000 species.
- Too large for our edge device.
- Provides the "expert knowledge" via soft labels.

Student: Our Model

- Will be trained to imitate BirdNet's predictions.

Benchmarking Against BirdNet

To objectively measure our model's performance against BirdNet, we need a controlled experiment.

1. **Common Test Set**
2. **"Apples-to-Apples" Metrics:** compare using identical metrics
   - Our **baseline model** vs. BirdNet
   - Our **distilled model** vs. BirdNet