CPU-Constrained Multi-Signal Pipeline for Bioacoustic Wildlife Monitoring: BirdCLEF+ 2026 Working Note

Meryem Altundal[1,*]

[1]Yıldız Technical University, Istanbul, Turkey

Abstract

We describe a machine learning system for acoustic species identification submitted to the BirdCLEF+ 2026 competition, which targets 234 wildlife species in passive acoustic monitoring recordings from Brazil's Pantanal wetlands. The central engineering challenge is an inference-time constraint of 90 minutes on CPU-only hardware, which prevents the use of large or slow models. Our solution fuses three complementary signal sources—a fine-tuned MLP head on Google PerchV2 embeddings, the PerchV2 built-in classifier for 158 species with known taxonomy overlap, and prototype-based cosine similarity derived from expert-labeled soundscapes—within a TFLite FP16 inference pipeline that processes all twelve 5-second segments of every test recording in approximately 19 minutes total. Additional post-processing includes GIS-informed site priors, taxonomy-aware smoothing, and silence suppression. Our final submission achieved a macro-averaged ROC-AUC of 0.766 on the public leaderboard and 0.76568 on the private leaderboard (rank 3368/4091). We report an iterative ablation history, document three engineering failures that each caused measurable score regressions, and describe the AI-assisted development workflow that enabled rapid iteration in a cloud-only environment.

Keywords

bioacoustics, passive acoustic monitoring, species identification, TFLite, PerchV2, BirdCLEF, Pantanal

1. Introduction

Passive acoustic monitoring (PAM) is emerging as one of the most scalable tools for biodiversity assessment, particularly in large and remote ecosystems where direct observation is impractical. The BirdCLEF+ competition series [1] operationalises this challenge as a machine learning task: given one-minute field recordings collected by autonomous recorders, predict the probability that each of 234 species—birds, amphibians, mammals, insects, and reptiles—is audible in each five-second window.

BirdCLEF+ 2026 [2] focuses on the Brazilian Pantanal, one of the world's largest tropical wetlands and a site of high conservation urgency. The competition dataset draws from two sources: (1) archival single-species recordings from Xeno-canto and iNaturalist (`train_audio`), and (2) longer passive recordings from the actual monitoring sites (`train_soundscapes`), a subset of which are annotated by expert ornithologists. Crucially, only the passive soundscapes share the spectral and acoustic conditions of the hidden test set, creating a substantial domain gap for models trained exclusively on archival material.

The submission format requires a self-contained Kaggle Notebook running without internet access, on CPU only, in under 90 minutes across approximately 600 one-minute recordings. This constraint rules out large transformer ensembles and demands careful engineering of both the model and the inference loop.

Our work makes three primary contributions: (i) an efficient TFLite-based inference pipeline enabling processing of all 12 segments per test file within the time budget; (ii) a multi-signal fusion scheme combining three independent evidence sources; and (iii) an honest analysis of three engineering failures—each causing measurable score regressions—that we believe will be instructive for future participants.

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany

*Corresponding author.

meryem.6.altundal@gmail.com (M. Altundal)

Positioning relative to top-performing systems. We do not claim state-of-the-art accuracy. Public discussion among top-ranked teams in this competition indicates that systems scoring above 0.95 macro-averaged ROC-AUC typically rely on (a) ONNX rather than TFLite runtimes, which appear to offer a further constant-factor speedup over our TFLite pipeline; (b) prototype/metric-learning heads (e.g. ProtoSSM style architectures) rather than a plain MLP; and (c) ensembles of three or more independently trained models blended via pre-computed score CSVs, rather than the single-model, three-signal fusion used here. Our system occupies a deliberately more modest point in this design space, prioritising engineering transparency and a fully reproducible single-model pipeline over leaderboard-maximising ensembling. We view the gap between our score (0.766/0.766) and the top tier (0.95+) as primarily attributable to the absence of (b) and (c), rather than to any single fixable defect in our approach—a distinction we think is useful for participants choosing where to invest limited development time in future PAM competitions.

2. Data and Preprocessing

The competition provides 234 target species, including both eBird-coded avian species and iNaturalist-coded non-avian taxa (insects identified only to sonotype level). All audio is resampled to 32 kHz and encoded as OGG. Test soundscapes are one minute long; predictions are required for each of the twelve non-overlapping 5-second segments.

The labeled subset of `train_soundscapes` provides 1,478 expert-annotated 5-second segments drawn from passive monitoring sites. A total of 251 distinct species appear in the annotations; of these, 75 fall within the 234 competition target classes and have at least one labeled segment, forming the basis for the per-species prototypes described in Section 4.3. The remaining annotated species lie outside the competition taxonomy and are discarded. These labeled soundscapes are the only training data that share the acoustic domain with the hidden test set—they were recorded by the same passive monitoring equipment at overlapping geographic sites.

Audio is loaded using `soundfile` (libsndfile), which is 3–5× faster than `librosa.load` for OGG decoding. Stereo files are downmixed to mono by averaging channels; files shorter than the target duration are zero-padded; files longer are truncated. No spectral features are computed during inference—PerchV2 accepts raw waveforms at 32 kHz.

3. System Architecture

Figure 1 summarises the full inference pipeline, from the raw audio segment through the three-way signal fusion to the post-processing chain described in Section 5.

3.1. Backbone: Google PerchV2

Our backbone is Google's PerchV2 bird vocalization classifier [3], version 8, accessed via the `bird-vocalization-classifier` Kaggle model. PerchV2 accepts a raw waveform segment and returns both a 1280-dimensional embedding and a classifier output over 2333 bird species. We use both outputs: the embedding feeds our trainable MLP head, while the classifier output provides a zero-training-cost baseline for 158 of our 234 target species whose eBird codes appear in the PerchV2 taxonomy.

3.2. MLP Classification Head

A lightweight MLP maps the 1280-dimensional PerchV2 embedding to 234 class logits, with architecture $1280 \to 512 \to 256 \to 234$:

- **Layer 1**: Linear(1280, 512), BatchNorm, ReLU, Dropout(0.3)
- **Layer 2**: Linear(512, 256), BatchNorm, ReLU, Dropout(0.2)

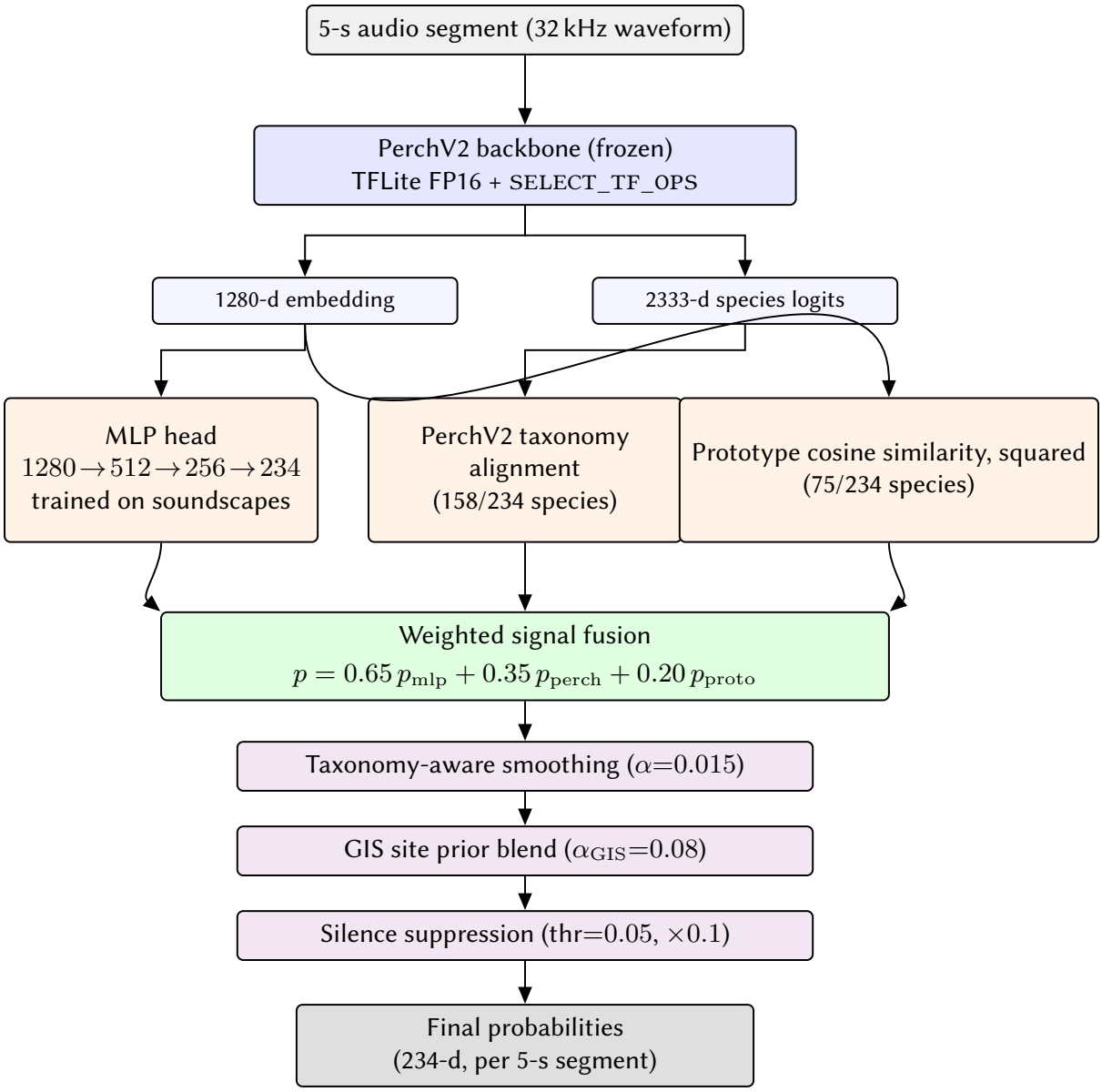

Figure 1: End-to-end inference pipeline. The PerchV2 backbone is frozen; only the MLP head (left branch) is trained. The PerchV2 taxonomy alignment and prototype cosine branches require no additional training and provide coverage for species subsets not well captured by the MLP alone.

- **Layer 3**: Linear(256, 234) — output logits

3.3. TFLite Optimisation for CPU Inference

PerchV2 is a TensorFlow SavedModel that cannot be used directly within a 90-minute CPU budget for 600 recordings. We convert it to TFLite FP16 format (≈25 MB) using the standard TFLite converter with SELECT_TF_OPS enabled—required because several PerchV2 operations are JAX-derived and lack native TFLite kernels. At inference time we load the TFLite interpreter with `num_threads=4` and the XNNPACK delegate, achieving approximately 0.16 s per 5-second segment. This enables processing of all $600 \times 12 = 7{,}200$ segments in ≈19 minutes. Without TFLite, runtime constraints forced us to process only 3 segments per recording (see Section 6.2). We did not perform a controlled comparison of full-precision versus FP16 accuracy on a held-out set; this is a limitation we flag explicitly, since isolating the effect of quantisation from the simultaneous removal of the `max_segs` truncation (Section 6.2) would require a dedicated ablation we did not have time to run before the submission deadline.

The interpreter exposes six output tensors; we identify the 1280-d embedding and 2333-d label tensors by their shapes at runtime.

4. Training Procedure

All trainable parameters reside in the MLP head; PerchV2 itself is not fine-tuned.

4.1. Data

Training uses exclusively the 1,478 expert-labeled `train_soundscapes` segments—not the archival `train_audio` files. We extract PerchV2 embeddings for every labeled segment in advance on the Kaggle GPU accelerator, cache them as `float32` arrays (shape 1478×1280), and train entirely in embedding space. This strategy directly targets the domain gap between archival recordings and field soundscapes. An earlier training run used 739 segments (before the full labeled soundscape release); the final run used all 1,478 available segments.

A site-based validation split (sites S22 and S08 held out) was defined during development, but the final model was trained on all available data without a held-out validation set, since the labeled soundscape dataset is small enough that withholding any portion measurably reduces training signal. Reported validation scores during training are therefore optimistic and should be interpreted accordingly.

4.2. Hyperparameters and Loss

- Epochs: 30; batch size: 64
- Optimiser: AdamW, lr $= 2 \times 10^{-5}$, weight decay 10^{-4}
- Scheduler: CosineAnnealingLR, $T_{\max} = 30$, $\eta_{\min} = 5 \times 10^{-7}$
- Loss: Focal BCE, $\alpha = 0.25$, $\gamma = 2.0$ [4]
- Augmentation: Gaussian noise ($\sigma = 0.01$) on embeddings; Mixup [5] with $\lambda \sim \text{Uniform}(0.1, 0.5)$
- Gradient clipping: $\|\nabla\|_\infty \leq 1.0$
- Initialisation: warm-started from best prior checkpoint

4.3. Prototype Extraction

After training, per-species prototype embeddings are computed by averaging the PerchV2 embeddings of all labeled soundscape segments for that species, then L2-normalising. This yields prototypes for 75 of the 234 target species; for the remaining 159 species, no labeled soundscape segments are available.

5. Multi-Signal Post-Processing Pipeline

Our inference pipeline fuses three evidence sources and applies four sequential post-processing steps.

Signal fusion. Let $\mathbf{p}_{\text{mlp}}$, $\mathbf{p}_{\text{perch}}$, and $\mathbf{p}_{\text{proto}}$ denote MLP sigmoid outputs, aligned PerchV2 classifier probabilities, and prototype similarity scores respectively. For each species j:

$$p_j = 0.65 \cdot p_{\text{mlp},j} + 0.35 \cdot p_{\text{perch},j} + 0.20 \cdot p_{\text{proto},j}$$

Weights are additive (not normalised to 1) to allow mild probability inflation for rare species. PerchV2 and prototype signals are applied only for the species they cover (158 and 75 species respectively).

Prototype cosine similarity. For prototype-covered species, similarity is sharpened by squaring: $p_{\text{proto},j} = \max(0,\, \hat{e}^\top c_j)^2$, where $\hat{e}$ is the L2-normalised segment embedding and c_j the L2-normalised species prototype.

Taxonomy-aware smoothing. Within each taxonomic class, probabilities are shifted toward the class mean with weight $\alpha = 0.015$, implementing a mild co-occurrence prior.

GIS site priors. The recording site code is extracted from the filename and used to blend predictions with a site-specific occurrence prior from training soundscape co-occurrence statistics ($\alpha_{\text{GIS}} = 0.08$).

Silence suppression. Segments where no species exceeds probability 0.05 have all predictions multiplied by 0.1 to reduce false positives in ambient-noise segments.

6. Failure Analysis and Lessons Learned

Three engineering mistakes each caused measurable leaderboard regressions. We document them in detail because the failure modes are subtle and likely to recur in future PAM competitions.

6.1. Temporal Filter UTC Offset Bug

To exploit diurnal/nocturnal activity patterns, we implemented a temporal filter that attenuated diurnal species (primarily Aves) during night hours and vice versa, using the UTC hour from the test filename.

A critical oversight: the Pantanal lies in the UTC-4 time zone. Bird dawn chorus occurs locally around 05:00–08:00, corresponding to 09:00–12:00 UTC. Our filter treated UTC 05–08 as morning (boost diurnal), which mapped to local 01:00–04:00—deep night—while the actual dawn chorus at UTC 09–12 received only neutral weighting. The result was that the most active dawn chorus recordings received diurnal attenuation rather than amplification.

Score impact: introducing this filter reduced public LB from 0.633 to 0.506. Score recovered to 0.766 after the filter was removed entirely.

Lesson: Time-based filters in PAM competitions must account for the local time zone of the recording site. We recommend explicitly converting UTC timestamps using the known UTC offset of the monitoring region before applying any circadian-based adjustments.

6.2. Silent Audio Truncation from `max_segs`

In an earlier notebook version (public LB: 0.633), inference was limited to `max_segs=3`: only the three highest-energy 5-second segments of each minute recording were processed. This was motivated by runtime concerns before TFLite was available.

This silently discarded 75% of each recording's audio. Species audible only in later portions of a recording—which the energy criterion failed to rank highly—received a predicted probability of zero. No error was raised; the output was simply wrong for a large fraction of segments. TFLite reduced per-segment latency to 0.16 s, enabling `max_segs=12` without exceeding the time budget.

Lesson: Throughput constraints must be addressed at the model level (quantisation, format conversion) rather than by silently truncating input. A truncation parameter that produces no error but drops most of the data is a particularly hazardous class of bug.

6.3. Architecture Mismatch in Warm-Starting

In an intermediate notebook iteration (public LB: 0.506), we widened the MLP hidden dimensions from $512 \to 256$ to $1024 \to 512 \to 256$, initialising with the trained narrow-network weights. We had implemented a fallback in which `load_state_dict` silently switched from `strict=True` to `strict=False` on shape mismatch.

With `strict=False`, only the layers with matching shapes were loaded; the expanded first layer was initialised to random values. The resulting model effectively started from a random initialisation in the first layer while inheriting calibrated later layers—a combination worse than either pure warm-starting or pure random initialisation.

Lesson: Silent `strict=False` fallbacks can corrupt warm starts. Validate the loaded parameter count and compute a forward pass on a small validation batch immediately after loading to confirm the model output is sensible before beginning training.

7. AI-Assisted Development Workflow

This project was conducted in a cloud-only, zero-local-environment setting: Kaggle Notebooks for GPU/CPU execution and Claude (Anthropic) as an AI coding assistant in a parallel browser tab—a "dual tab" workflow.

The role of Claude extended substantially beyond code autocompletion. Specific contributions included: (1) generating complete notebook cells for novel components, such as TFLite interpreter setup with runtime tensor-shape probing; (2) diagnosing multi-cell error cascades from pasted execution logs, identifying root causes not apparent from Python tracebacks (including the architecture mismatch described in Section 6.3); (3) helping design and implement the multi-signal fusion and the prototype similarity squaring nonlinearity; (4) identifying the UTC offset as the likely cause of the temporal filter regression from a symptom description alone; and (5) managing the constraint that training and inference must reside in separate Kaggle artefacts due to the GPU/CPU split.

The interaction pattern was iterative: a complete error log or failing cell was shared with Claude, which produced a corrected replacement with an explicit placement instruction. Debugging cycles typically completed in under five minutes. The UTC offset bug (Section 6.1) was identified within minutes of describing the symptom—a diagnosis that would otherwise have required manual timestamp inspection and timezone documentation lookup.

We believe transparent reporting of AI-assisted workflows is valuable for the community to understand how these tools change the experimental iteration rate and the nature of the errors that are caught (or missed).

8. Results

Table 1 summarises public leaderboard scores at each iteration. The final system achieved 0.76568 on the private leaderboard (rank 3368/4091). The near-parity between public (0.766) and private (0.76568) scores suggests reasonable generalisation across the two test splits.

Table 1
Public leaderboard score progression (macro-averaged ROC-AUC).

Version	Key change	Public AUC
v1	EfficientNet-B0 baseline, 206 classes	0.509
v2	PerchV2 + MLP + GIS prior + temporal filter (UTC bug)	0.633
v3	Wider MLP (1024-512-256) + architecture mismatch	0.506
v4 (final)	TFLite FP16, all 12 segs, prototype, filter removed	**0.766**

Private LB (final): 0.76568 (rank 3368/4091)

The gap between our final score (0.766) and the top-tier systems (>0.95) is primarily attributable to: (1) use of a single model rather than an ensemble; (2) no fine-tuning of the PerchV2 backbone; and (3) no pseudo-labeling of unlabeled soundscapes.

9. Conclusion

We presented a CPU-efficient multi-signal pipeline for acoustic species identification in BirdCLEF+ 2026. The central engineering contribution is TFLite FP16 conversion of PerchV2 with SELECT_TF_OPS, enabling processing of all 12 test segments per recording within the 90-minute CPU budget. A three-way

signal fusion combining MLP predictions, PerchV2 built-in classifier outputs, and prototype cosine similarity provides complementary species coverage.

Equally important, we document three engineering failures that together demonstrate a class of bugs common in constrained, cloud-only ML settings: bugs that produce no error message but silently degrade output quality. The UTC offset temporal filter, the `max_segs` truncation, and the silent `strict=False` warm-start failure all share this property. We hope this analysis is useful to future BirdCLEF participants.

Reproducibility. The final inference notebook (public leaderboard score 0.766) is publicly available on Kaggle.[1] An open-source release of the standalone TFLite pipeline and a browser-based demo are planned as follow-on work.

Future work would prioritise: pseudo-labeling of unlabeled soundscapes to extend prototype coverage, fine-tuning of later PerchV2 layers with a very low learning rate, and construction of a second independent model for ensemble blending.

Declaration on Generative AI

During the preparation of this work, the author used Claude (Anthropic) in order to: generate and debug notebook code, diagnose multi-cell error cascades from execution logs, help design and implement components of the post-processing pipeline, and assist with manuscript drafting and LaTeX formatting. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] L. Picek, L. Adam, S. Kahl, R. Bossy, L. Chrobak, H. Goëau, K. Papafitsoros, H. Klinck, W.-P. Vellinga, R. Planqué, T. Denton, K. Barnard, C. Nédellec, L. Deléger, M. Courtin, G. Martellucci, I. Moummad, F. Vinatier, P. Bonnet, A. Joly, Overview of LifeCLEF 2026: Ai challenges for biodiversity understanding and ecosystem management, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2026.

[2] S. Kahl, T. Denton, L. Sugai, L. Piatti, W. H. Arruda Moreno, M. M. Barbosa, M. Eibl, C. Z. Fieker, C. M. Garcia, D. L. Hokama Sousa, J. E. d. A. Júnior, R. C. Kridler, M. Lasseck, A. V. d. Melo, H. Müller, M. d. O. Neves, M. G. d. Reis, L. K. Sampaio Teles, K.-L. Schuchmann, J. L. M. M. Sugai, K. T. P. Azevedo, P. d. N. Lopes, M. I. Marques, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2026: Acoustic species identification in the Pantanal, South America, in: Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, 2026.

[3] B. Ghani, T. Denton, S. Kahl, H. Klinck, Global birdsong embeddings enable superior transfer learning for bioacoustic classification, Scientific Reports 13 (2023) 22876. doi:10.1038/s41598-023-49989-z.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[5] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations (ICLR), 2018.

[1] https://www.kaggle.com/code/meryemaltundal/birdclef08-inference