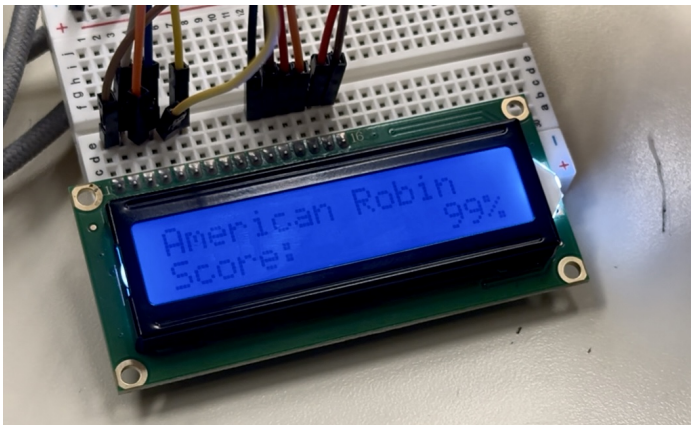
PocketBirdNET: Knowledge Distillation for Data-Efficient On-Device Bird-Sound Classification

Rishabh Goenka Aarav Wadhwani Carson Large

Department of Electrical and Computer Engineering, University of Washington

June 2026



The 60.3 KB classifier running offline on an Arduino Nano 33 BLE Sense.

Abstract. State-of-the-art bird-sound recognizers such as BirdNET (∼50 MB, 6,522 classes) require a phone or cloud connection, ruling them out for offline, milliwatt-scale field monitoring. We ask whether such a model can be distilled onto a microcontroller without losing usable accuracy, and—more interestingly—what knowledge distillation actually *buys* once it is. Training one fixed depthwise-separable CNN four ways on identical data (from scratch, ImageNet transfer, distillation from BirdNET, and a distilled-then-INT8 deployment model), we find that distillation primarily buys *data efficiency*: a from-scratch network collapses to chance until it has a large labeled set, whereas the distilled network is already useful with a fraction of the data, and the gap closes only as data grows. The deployed INT8 model is 60.3 KB, fits an 84.8 KB tensor arena on a Cortex-M4F, and retains essentially all of its float accuracy (macro-F1 0.634 vs. 0.635). Results are delivered both on an on-device LCD and over BLE to a web dashboard.

Code: `https://github.com/rishabhlgoenka/PocketBirdNET`
Live dashboard: `https://pocketbirdnet.vercel.app/`

1 Introduction

Passive acoustic monitoring—deploying microphones in the field and classifying the recorded audio—has become a standard tool for tracking bird populations. The recognition models that make it work, however, are large: BirdNET [1], the de-facto open recognizer, is roughly 50 MB and covers 6,522 classes, and running it implies a phone or a network link. That is a poor fit for the deployment that motivates this work: remote, battery-powered sensors that must run for weeks with no connectivity and a milliwatt power budget. TinyML—inference on a microcontroller—removes the network dependence entirely, but only if a usable model fits in a few hundred kilobytes of RAM.

Knowledge distillation [2] is the natural tool: train a small "student" to imitate a large "teacher," transferring the teacher's learned structure into a model small enough to deploy. The usual framing treats distillation as a way to recover accuracy lost by shrinking a model. We are interested in a sharper question: holding the student architecture and the data fixed, *what does distilling from BirdNET actually give us* relative to simply training the same network from scratch or from a generic pretrained backbone?

To answer this we train one fixed depthwise-separable CNN four ways on an 11-class task (10 Pacific Northwest species plus a background class) and compare them on identical data and evaluation: (A) from scratch, (B) ImageNet transfer learning, (C) knowledge distillation from BirdNET, and (D) the distilled model after INT8 post-training quantization, which is the version actually flashed to hardware.

Our findings are:

- **Distillation buys data efficiency.** A from-scratch network sits at chance until it is given a large labeled set; the distilled network is already useful with a fraction of that data, and the advantage is largest precisely in the low-data regime that field deployment lives in.
- **Compression to the edge is nearly free.** INT8 quantization shrinks the distilled model 2.4× to 60.3 KB—inside the RAM and flash budget of a Cortex-M4F—for a 0.001 change in macro-F1.
- **Generic transfer does not help here.** ImageNet features transfer essentially nothing to mel-spectrograms; variant B slightly *underperforms* training from scratch.

2 Methods

2.1 Dataset and preprocessing

Audio was drawn from the Xeno-Canto archive [4] for ten Pacific Northwest species—American Robin, Black-capped Chickadee, Steller's Jay, Northern Flicker, Song Sparrow, Anna's Hummingbird, Dark-eyed Junco, American Crow, Pacific Wren, House Finch—plus an eleventh *background* class of ambient, no-bird audio. Queries were filtered to quality grades A–B and 10–120 s clips. In total 3,546 recordings yielded 75,027 raw three-second windows; per-class audio ranges from 163 min (Anna's Hummingbird) to 460 min (American Robin).

Each window is resampled to 16 kHz mono and converted to a 40×32 log-mel spectrogram (2,048-sample Hann window, 1,500-sample hop, 40 mel bins over 125–7,500 Hz), normalized to $[0, 1]$. The identical feature routine is mirrored bit-for-bit in the on-device C frontend, so a model that works in Python behaves identically on hardware. Data are split 70/15/15 *at the recording level*, stratified by species, so that no recording contributes windows to more than one split; this prevents the temporal leakage that would otherwise inflate test accuracy. Training windows (and only training windows) are augmented with time shift, Gaussian noise, frequency masking, and mixup with background, giving 104,294 training examples. Fig. 1 shows one example per class.

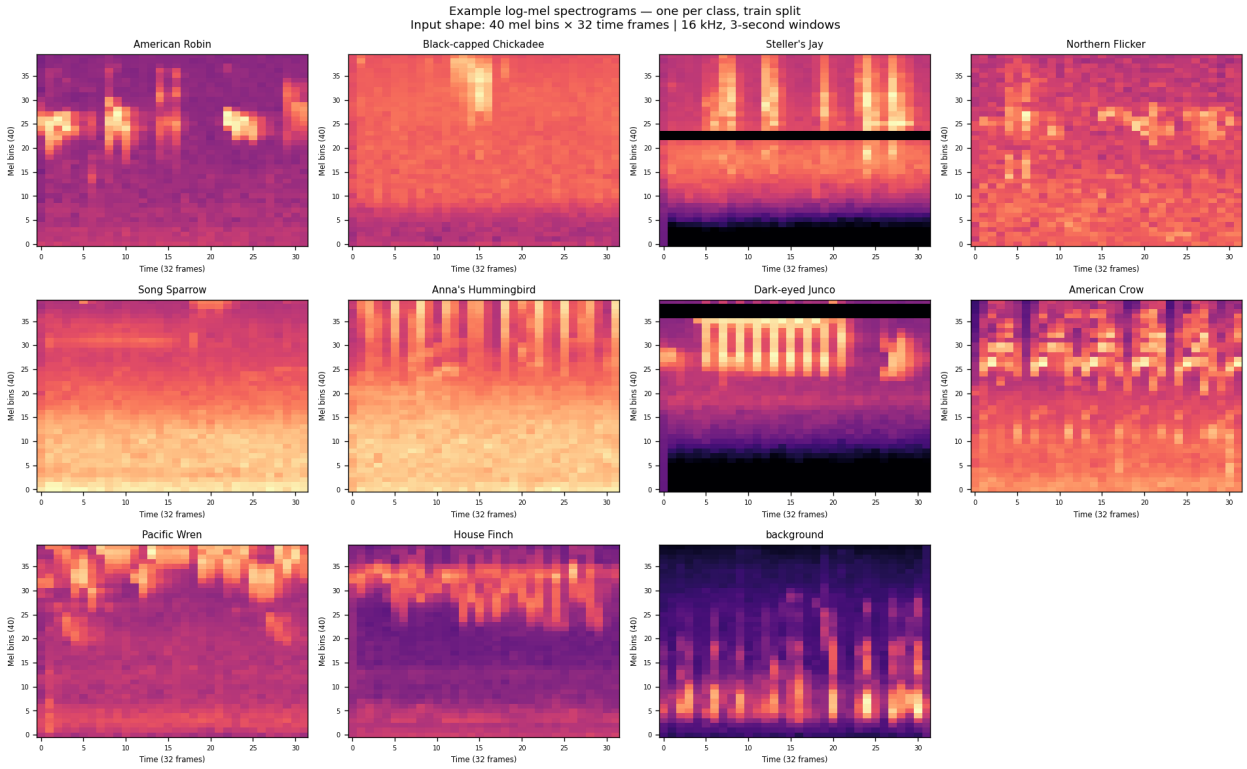


Figure 1. Log-mel inputs (40×32, 3 s at 16 kHz), one per class. Horizontal gaps are the frequency-mask augmentation.

2.2 Student architecture

All four variants share one network: a depthwise-separable CNN (width multiplier $\alpha = 0.5$) with a 16-filter stem, four depthwise-separable blocks (32, 64, 128, 128 channels, stride-2), global average pooling, a 32-unit dense layer with dropout, and an 11-way softmax. It has 36,363 parameters (142 KB in float32, 60.3 KB in INT8). Fixing the architecture across variants is what isolates the *training signal* as the only independent variable.

2.3 Training variants and distillation

Variant A trains on hard labels from random initialization. Variant B fine-tunes an ImageNet-pretrained MobileNetV2 [3] backbone (the $40 \times 32 \times 1$ input is resized and channel-replicated). Variant C distills from BirdNET: BirdNET is run at its native 48 kHz over the training recordings, its species confidences are converted to pseudo-logits, and the student minimizes

$$\mathcal{L} = \alpha\, T^2\, \mathrm{KL}\big(\sigma(z_t/T) \,\|\, \sigma(z_s/T)\big) + (1-\alpha)\, \mathrm{CE}(y, z_s), \tag{1}$$

with temperature $T = 4$. One detail mattered: BirdNET emits a large background logit whenever it is uncertain—even on genuine bird windows—so we mask the background dimension before the teacher softmax, letting the soft target carry only the 10-species distribution while the hard-label term handles background membership. Without this mask the student collapses to predicting background everywhere. BirdNET's standalone top-1 agreement with ground truth on our windows is only 45.7%, consistent with reviews reporting that its accuracy is strongly species- and context-dependent [6]; it is thus a useful but imperfect teacher, not a near-oracle. Variant D is variant C after INT8 post-training quantization. Optimizer, schedule, and the α sweep that selected $\alpha = 0.1$ are given in Appendix A.

2.4 Deployment pipeline

The float model is converted to a fully integer (INT8 in/out) TFLite model with a 256-sample calibration set, emitted as a C array, and compiled into firmware running on LiteRT for Microcontrollers [5]. On the device, the PDM microphone feeds a ring buffer; the C frontend computes the mel spectrogram incrementally (avoiding a 96 KB raw-audio buffer); the INT8 network runs; and a confidence threshold gates the output, which is shown on a 16×2 LCD and streamed as a 2-byte BLE packet to a web dashboard. Fig. 2 shows the full pipeline.

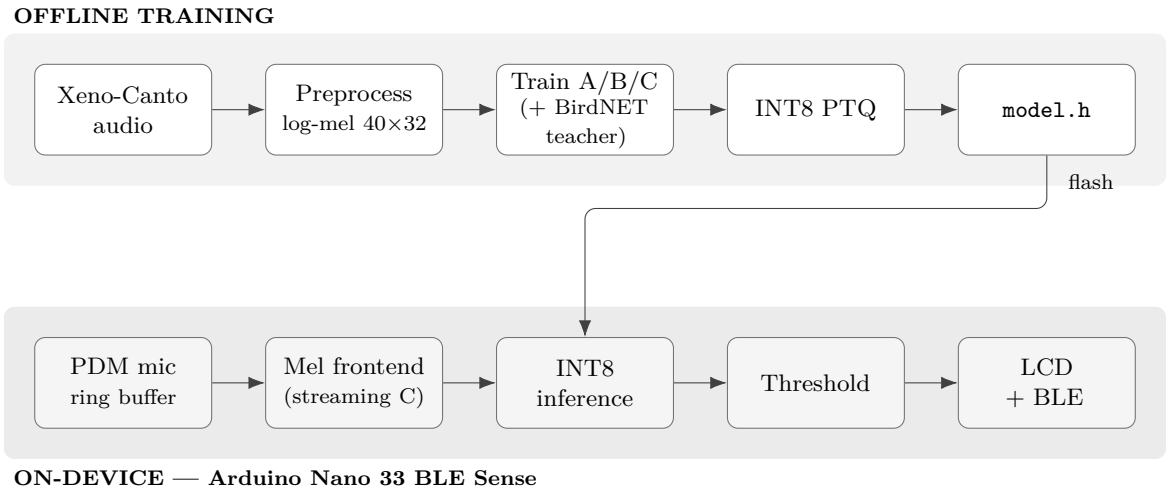


Figure 2. End-to-end pipeline: offline training and INT8 conversion (top); on-device inference on the Cortex-M4F with LCD and BLE output (bottom).

3 Results

3.1 Distillation buys data efficiency

The central result comes from training the same pipeline at three increasing data volumes (Fig. 3, with exact values in Table 1). From-scratch training (A) is the revealing case: it sits at the 11-class chance level (macro-F1 ≈ 0.015) at both small and medium data, then jumps to 0.609 only once the full dataset is available. The architecture is therefore not the bottleneck—the same network reaches competitive accuracy when given enough labels. What changes is how many labels each strategy needs. Distillation (C) is already useful at the smallest scale (0.43 macro-F1, vs. chance for scratch) and leads at every scale; the deployed INT8 model (D) tracks it throughout. The gap between scratch and the distilled models shrinks monotonically as data grows—the signature of a teacher supplying structure that the student would otherwise have to discover from far more examples. For a focused field deployment, where labeled audio for the target species is the scarce resource, this is the property that matters most.

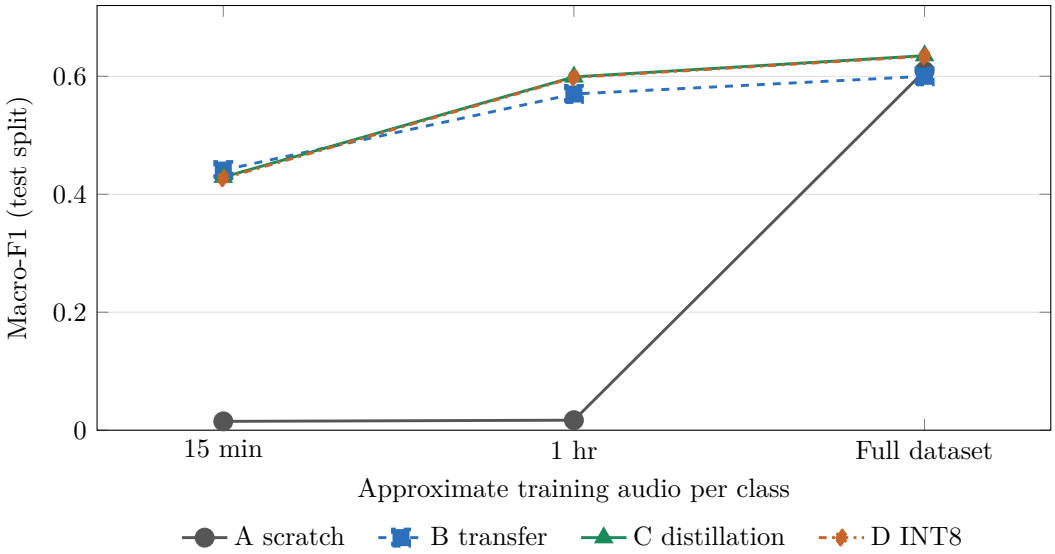


Figure 3. Macro-F1 versus training-data volume. Scratch (A) is at chance until the full dataset, then climbs sharply; the distilled models (C, D) lead at every scale, most strongly when data is scarce. Exact values in Table 1.

Table 1. Macro-F1 by training-data volume (companion to Fig. 3).

Variant	∼15 min	∼1 hr	Full dataset
A — Scratch	0.015	0.017	0.609
B — Transfer	0.441	0.570	0.600
C — Distillation	0.429	0.599	**0.635**
D — INT8 deploy	0.427	0.598	0.634

3.2 Four-way comparison on the full dataset

On the full dataset (Table 2), distillation (C) is the most accurate variant at 0.635 macro-F1, and the deployed INT8 model (D) retains 0.634 while being the only variant small enough to run on the target hardware. Transfer learning (B) lands slightly below from-scratch training, which we return to in Section 4.

Table 2. Test-split comparison on the full dataset. Sizes for A/B/C are float32 parameter footprints (not deployable); D is the INT8 model flashed to the board.

ID	Description	Top-1	Macro-F1	Size
A	Scratch	61.9%	0.609	142 KB (f32)
B	Transfer (MobileNetV2)	61.0%	0.600	9,143 KB (f32)
C	Distillation (T=4)	64.4%	0.635	142 KB (f32)
D	Distilled + INT8 (deployed)	64.2%	**0.634**	**60.3 KB**

3.3 Per-class behavior

Per-class F1 (Table 3) is uneven in interpretable ways. Acoustically distinctive species—American Robin, Pacific Wren, House Finch—and the background class exceed 0.70, while the highly variable Song Sparrow is weakest at ~ 0.52. The largest distillation gains appear where they should: Anna's Hummingbird, the class with the fewest training windows, and American Crow (+0.11 F1 over

transfer), again pointing to data efficiency. The confusion matrices (Fig. 4) show clean diagonals for the deployed model, with residual confusion concentrated among acoustically similar species.

Table 3. Per-class F1 on the full dataset (test split). Best per row in bold.

Class	A	B	C	D
American Robin	0.686	0.675	**0.730**	0.721
Black-capped Chickadee	**0.656**	0.614	0.627	0.621
Steller's Jay	**0.568**	0.538	**0.568**	0.558
Northern Flicker	0.526	0.515	0.532	**0.535**
Song Sparrow	0.500	0.508	0.520	**0.523**
Anna's Hummingbird	0.509	0.517	**0.532**	**0.532**
Dark-eyed Junco	0.646	0.622	**0.662**	0.661
American Crow	0.585	0.519	0.629	**0.632**
Pacific Wren	0.667	0.679	0.740	**0.745**
House Finch	0.645	0.667	0.699	**0.707**
background	0.709	**0.742**	0.748	0.744
Macro avg	0.609	0.600	**0.635**	0.634

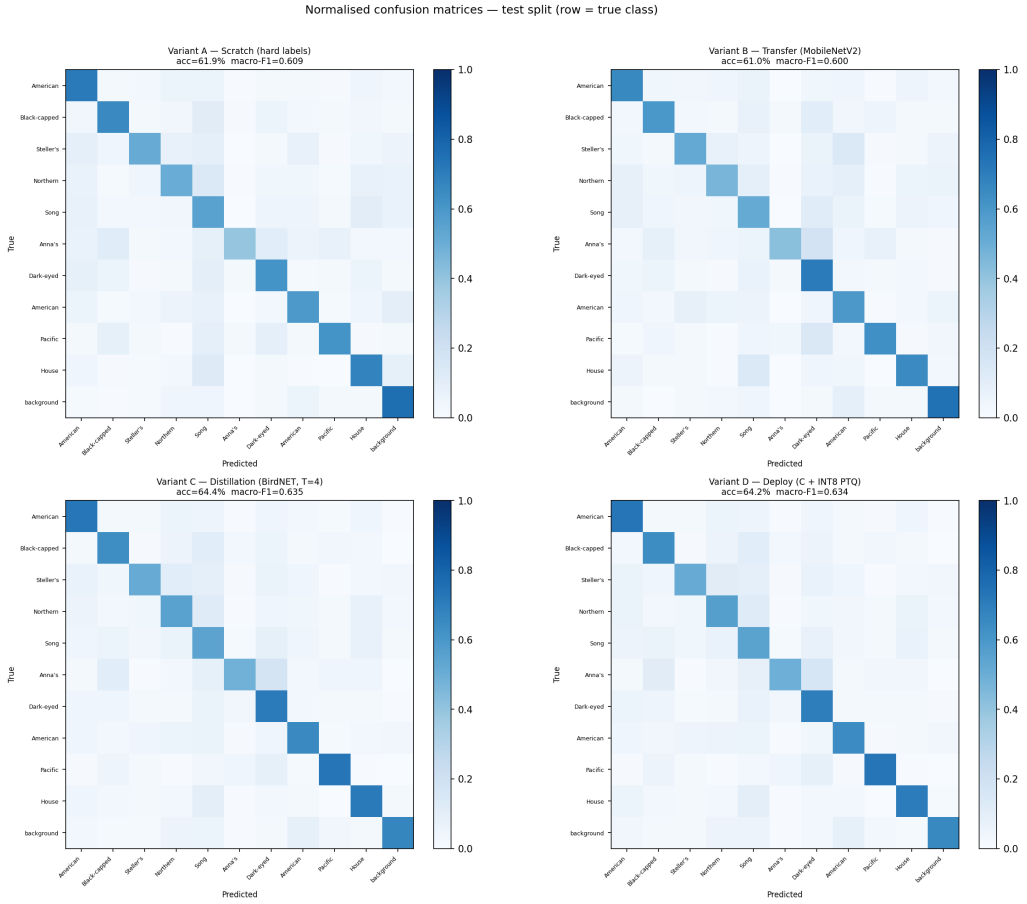


Figure 4. Normalized confusion matrices (test split, rows = true class) for variants A–D on the full dataset.

3.4 Compression and on-device deployment

Quantizing the distilled model to INT8 is effectively lossless here: a 2.4× size reduction (142 KB → 60.3 KB) for a 0.001 macro-F1 change, so quantization-aware training was unnecessary. The deployed model uses an 84.8 KB tensor arena; with the mel-frontend scratch and audio buffers it occupies roughly 131 KB of the 256 KB SRAM, leaving real headroom. End-to-end response is about 1.2 s after the 3 s capture window, dominated by the on-device mel frontend. In live testing on the onboard microphone, played calls across all ten species and background were classified correctly, with predictions shown on the LCD and mirrored on the dashboard (Fig. 5).

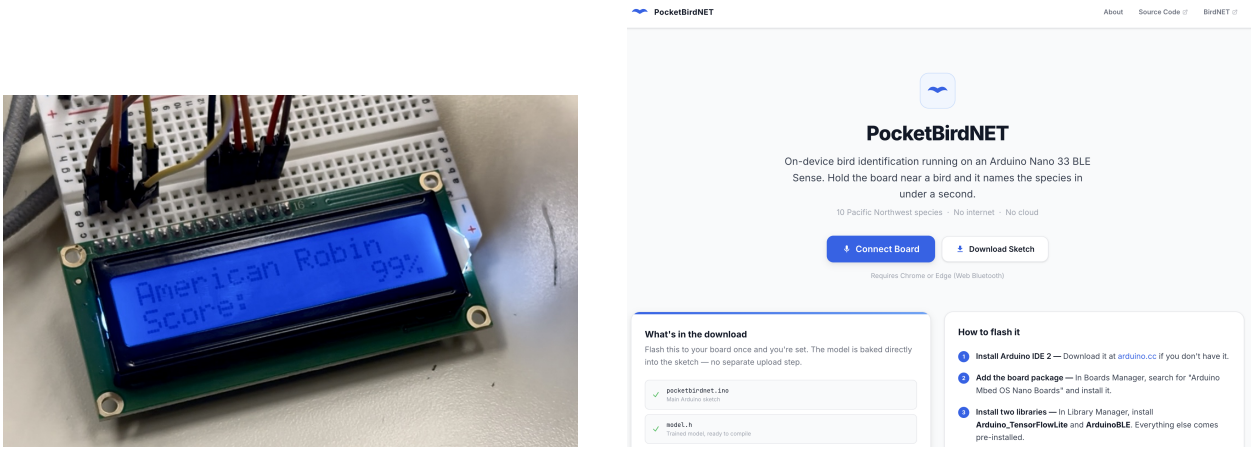


Figure 5. Deployed system. Left: the Nano 33 BLE Sense with the LCD showing a live classification. Right: the web dashboard receiving a detection over BLE.

4 Discussion

The data-scaling result reframes what distillation is doing here. It is not merely recovering accuracy lost to a small architecture—the from-scratch model proves the architecture can reach 0.61 on its own given enough data. Rather, the BirdNET teacher injects bird-acoustic structure that a randomly initialized network would otherwise have to recover from many more labeled examples, so the distilled student climbs the data curve far faster. This is the practically relevant behavior for the target application: in a focused deployment, curated labeled audio for the specific species of interest is the bottleneck, and distillation is most valuable exactly there.

Two negative results are as informative as the positive one. First, transfer from ImageNet (B) does not help and slightly hurts: features tuned for natural-image edges and textures do not align with mel-spectrogram structure, and the input adaptation adds parameters without matching inductive bias—a reminder that "transfer learning" is only useful when the source domain shares structure with the target. Second, naive distillation initially failed by collapsing to the background class, because the teacher expresses uncertainty as high background confidence; the fix—masking the teacher's background logit so the soft target is purely about *which species*—was what made distillation work at all, and is the kind of teacher-signal hygiene that matters whenever the teacher's label space and confidence behavior differ from the student's.

5 Limitations and Future Work

Absolute accuracy (∼64%) is modest; the scaling study attributes this to data volume rather than architecture, and more curated per-class audio is the most direct lever. All training audio is clean Xeno-Canto material, so a domain gap remains to the noisier onboard microphone; collecting and fine-tuning on field-recorded PDM audio is the natural next step. We did not deploy pruning

(the toolchain could not prune the functional model and the size budget was already met) and did not explore quantization-aware training or sub-INT8 formats, both of which could extend the compression frontier. Finally, the model is single-label per window; overlapping calls would motivate a multi-label head.

6 Conclusion

A state-of-the-art bird recognizer can be distilled into a 60.3 KB INT8 network that runs fully offline on a Cortex-M4F microcontroller with negligible accuracy loss relative to its float counterpart. More importantly, a controlled four-way comparison on fixed data and architecture shows that knowledge distillation's main contribution in this setting is *data efficiency*: it makes a deployable model useful with far less labeled audio than from-scratch training requires, with the largest advantage in the low-data regime that real field deployments inhabit.

References

[1] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "BirdNET: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, 101236, 2021.

[2] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.

[4] Xeno-Canto Foundation, "Xeno-canto: Sharing bird sounds from around the world," `https://www.xeno-canto.org` (API v3).

[5] TensorFlow Authors, "LiteRT for Microcontrollers," `https://ai.google.dev/edge/litert/microcontrollers/overview`.

[6] C. Pérez-Granados, "BirdNET: applications, performance, pitfalls and future opportunities," *Ibis*, 2023, doi:10.1111/ibi.13193.

A Implementation details

Training used Adam (LR 10^{-3}), batch size 64, up to 80 epochs with early stopping on validation loss (patience 12) and ReduceLROnPlateau ($\times 0.5$, patience 6, floor 10^{-6}), seed 42, in TensorFlow 2.21 / Keras 3.14. Variant B fine-tuned in two phases (head-only, then the last 30 backbone layers at LR 5×10^{-5}, input resized to 96×96). The distillation weight was selected by sweep: $\alpha = 0.1$ gave validation macro-F1 0.660, vs. 0.642 at 0.3 and 0.627 at 0.5. INT8 conversion used a 256-sample calibration set (input scale 0.003922, zero-point -128); the five model operators (`CONV_2D`, `DEPTHWISE_CONV_2D`, `FULLY_CONNECTED`, `MEAN`, `SOFTMAX`) are all in the standard LiteRT-for-Microcontrollers resolver. BLE uses a custom service/characteristic (`19b10000-...` / `19b10001-...`, `BLERead | BLENotify`) carrying a 2-byte `[classIndex, confidence]` payload; display thresholds are $\geq 70\%$ (BLE) and $\geq 60\%$ (LCD). Full code and reproduction instructions are at `https://github.com/rishabhlgoenka/PocketBirdNET`.