HARDWARE-AWARE MOBILENETV2-LITE WITH TEACHER-POOL DISTILLATION FOR BIODCASE-TINY 2026 TASK 3

Technical Report

Xin Hai[1], *Menghang Yin*[1], *Xingyue Li*[1], *Jinzhe Wang*[1]
Gongping Huang[1], *Yuzhu Wang*[2]

[1]School of Electronic Information, Wuhan University, Wuhan, Hubei, China
[2]Signal Processing Research Center, Tampere University, Tampere, Finland
hxpeckish@gmail.com

ABSTRACT

BioDCASE-Tiny 2026 Task 3 evaluates bird-sound recognition systems under embedded-device constraints. We propose a hardware-aware MobileNetV2-lite system for the ESP32-S3 platform. Instead of selecting a compact model only by parameter count, we design the student network around operator types that are well supported by the TensorFlow Lite Micro and ESP-NN deployment path, including int8 Conv2D, pointwise convolution, depthwise convolution, residual addition, global average and max pooling, and fully connected layers. The acoustic frontend follows the official 40-bin log-mel configuration without modification.

To improve the compact student's recognition ability, we distill soft targets from a 19-model teacher pool. The teachers are complementary in frontend resolution, convolutional architecture, temporal-frequency modeling, pooling strategy, regularization, and random seed. Their averaged predictions reached about 79% validation accuracy and were used as sample-aligned soft labels for the deployable student.

On the official validation split, our final int8 MobileNetV2-lite model achieved 0.7455 accuracy and 0.9470 ROC-AUC. On ESP32-S3-Korvo-2, the submitted TFLite Micro model used a 78.4 KB TFLite file, 5.38M MACs, 95.9 KB tensor arena, and 92.7 ms model inference time. Compared with the official embedded baseline, the system improved accuracy from 0.5628 to 0.7455 while reducing MACs, model size, memory usage, and board inference time.

***Index Terms*—** BioDCASE-Tiny, bird sound classification, TinyML, ESP32-S3, MobileNetV2, knowledge distillation

1. INTRODUCTION

BioDCASE-Tiny 2026 Task 3 [1] extends the recent BioDCASE tiny-hardware benchmark setting [2] and requires a bird-sound classifier that is accurate enough for acoustic recognition and small enough for microcontroller deployment. In this setting, model quality cannot be judged only by validation accuracy or parameter count. The final system must also use operators that are available in the embedded runtime, quantize reliably to int8, fit the tensor arena, and execute within a practical latency budget.

Our system treats the ESP32-S3 as a concrete deployment target. The ESP32-S3 software stack provides optimized paths for common int8 convolutional kernels through TensorFlow Lite Micro and ESP-NN [3]. We therefore used hardware compatibility as an architectural constraint and selected a MobileNetV2-lite student whose main computation consists of Conv2D, depthwise convolution, pointwise convolution, residual Add, global average/max pooling, concatenation, Dense, and Softmax operators. This graph avoids sequence models and attention modules at inference time, reducing the risk of unsupported or inefficient embedded execution.

Accuracy is improved by teacher-pool knowledge distillation. A pool of 19 stronger teachers provides averaged soft class probabilities, while the deployed student keeps the official low-cost frontend and the same lightweight inference graph. This separates training-time supervision from deployment-time complexity: the teacher pool is used only to produce targets, and the final model remains a single compact int8 TFLite Micro network.

2. PROPOSED SYSTEM

2.1. Official Frontend

The acoustic frontend is kept identical to the official Task 3 frontend. Audio is resampled to 24 kHz and represented as a 40-bin log-mel spectrogram over a 3 s clip. The frontend uses a 4096-sample analysis window, a 512-sample hop, and a 125–7500 Hz mel range. After transposition and channel insertion, the neural network input shape is $40 \times 133 \times 1$.

Keeping the frontend unchanged has two practical advantages. First, it preserves compatibility with the official evaluation and embedded code path. Second, it avoids moving computation from the neural network into a more expensive or less portable signal-processing frontend. The accuracy gain therefore comes from the student architecture and teacher supervision rather than from changing the input representation.

2.2. ESP32-S3-Oriented Model Design

The student model is a MobileNetV2-lite network designed around ESP32-S3-friendly int8 operations, following the MobileNet line of efficient mobile architectures [4, 5]. The model begins with a compact Conv2D stem, followed by lightweight inverted-bottleneck blocks. Each block uses pointwise channel expansion, 3×3 depthwise convolution, pointwise projection, and residual addition when tensor shapes match. The final feature map is aggregated by parallel global average and max pooling, concatenated, and classified by a small dense head.

Table 1: ESP32-S3 operator cues used in the student design.

Operator cue	Use in our student
Conv2D SIMD kernels	Stem and downsampling on regular int8 shapes.
1×1 pointwise Conv	Channel mixing with ESP-NN Conv coverage.
3×3 DepthwiseConv2D	Main local time-frequency modeling block.
Residual Add	Kept only for matched-shape lightweight skips.
Global average/max pooling	Head computation on the final small feature map.
RNN / attention avoided	Removes unsupported-op and slow-path risk.

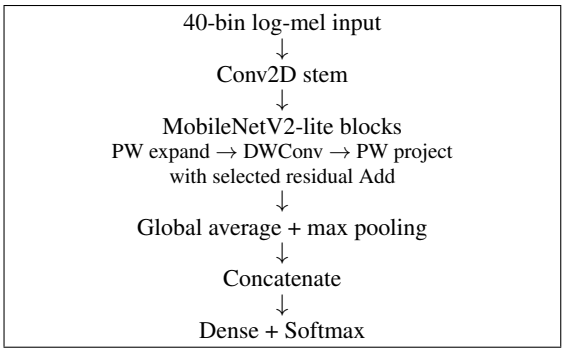

Figure 1: Deployable MobileNetV2-lite student graph.

Table 1 summarizes the operator-level cues used for the student design. These cues are not treated as fixed latency predictions; they indicate which graph patterns are most likely to reach the optimized ESP32-S3 execution path.

Figure 1 shows the inference-time graph. The graph contains only the operations required by the final deployed model; teacher models are not part of inference.

3. TEACHER-POOL DISTILLATION

The student is trained with soft targets from a 19-model teacher pool. The teacher pool is designed to be diverse rather than homogeneous. Its members cover different frontend resolutions, convolutional backbones, time-frequency modeling choices, pooling heads, regularization settings, and random seeds. In particular, the teachers may use higher-resolution acoustic frontends than the deployed 40-bin student, so distillation transfers information from richer training-time representations without increasing inference cost. The averaged teacher prediction reached approximately 79% validation accuracy and provided stronger class-probability structure than one-hot labels alone.

For an input sample x_i with hard label y_i, let $z_s(x_i)$ be the student logits and $z_m(x_i)$ be the logits of teacher m. The teacher-pool soft target is

$$\bar{p}_t(x_i;T) = \frac{1}{M}\sum_{m=1}^{M} \mathrm{softmax}\left(\frac{z_m(x_i)}{T}\right), \tag{1}$$

Table 2: Main training and deployment settings for the submitted system.

Item	Setting
Input	3 s audio, 24 kHz, $40 \times 133 \times 1$ features
Optimizer	Adam
Batch size	32
Learning rate	10^{-3}
Epochs	100, with early stopping patience 8
Distillation	19-teacher averaged soft targets, $T = 2$, hard/soft weights 0.3/0.7
Quantization	Full-int8 TFLite export with representative calibration
Board test	ESP32-S3-Korvo-2, ESP-IDF v5.4.4, 240 MHz CPU

where $M = 19$ and T is the distillation temperature. The student distribution is

$$p_s(x_i;T) = \mathrm{softmax}\left(\frac{z_s(x_i)}{T}\right). \tag{2}$$

The training objective combines hard-label cross entropy and soft-target distillation:

$$\begin{aligned}\mathcal{L} = {} & \lambda\,\mathrm{CE}\left(y_i, p_s(x_i;1)\right) \\ & + (1-\lambda)T^2\,\mathrm{KL}\left(\bar{p}_t(x_i;T)\,\|\,p_s(x_i;T)\right).\end{aligned} \tag{3}$$

In the final system, $T = 2$ and $\lambda = 0.3$. Thus, the teacher pool carries the larger supervision weight, while the hard label keeps the optimization anchored to the annotated class.

4. IMPLEMENTATION AND DEPLOYMENT

Table 2 gives the main implementation settings. The student was optimized with Adam. Training used a maximum of 100 epochs with early stopping patience of 8 epochs; the best validation checkpoint was exported rather than the last epoch. SpecAugment-style time-frequency masking was used during model development. The exported TFLite model was fully int8 quantized with representative calibration and then tested on the ESP32-S3-Korvo-2 board.

For embedded deployment, the generated source uses a slim operator resolver rather than an all-op resolver. The submitted source also includes the same runtime path used in local board testing, including the Conv2D channel-dimension fix required by the tested ESP-NN/TFLite Micro integration. This makes the submitted source code, model file, and board-tested runtime path consistent.

5. RESULTS

Table 3 compares the proposed system with the official embedded baseline. On the official validation split [6], the proposed int8 model achieved 0.7455 accuracy and 0.9470 ROC-AUC, compared with 0.5628 and 0.8923 for the baseline. At the same time, it reduced the TFLite model size from 111.4 KB to 78.4 KB, MACs from 23.32M to 5.38M, tensor arena usage from 202.6 KB to 95.9 KB, and model inference time from 330.36 ms to 92.72 ms.

The board profiler shows that most inference time is spent in convolutional operators, as expected for the selected architecture. On the tested ESP32-S3-Korvo-2 setup, Conv2D operators took

Table 3: Validation and embedded results.

Metric	Official baseline	Proposed
Validation accuracy	0.5628	0.7455
ROC-AUC	0.8923	0.9470
TFLite size (KB)	111.4	78.4
MACs (M)	23.32	5.38
Tensor arena (KB)	202.6	95.9
Model inference (ms)	330.36	92.72

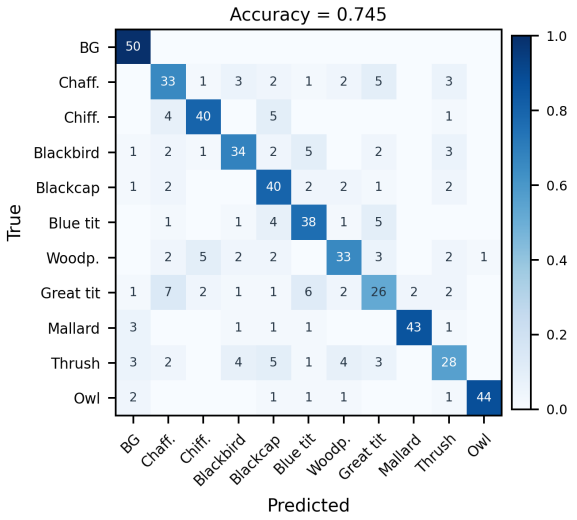

Figure 2: MobileNetV2-lite confusion matrix.

46.76 ms in total and DepthwiseConv2D operators took 29.07 ms. The remaining operations were smaller contributors: residual Add took 4.75 ms, global mean pooling 5.86 ms, global max pooling 5.43 ms, fully connected layers 0.65 ms, and Softmax 0.15 ms. The model ran without unsupported operators, tensor allocation failure, watchdog reset, abort, or crash.

6. CONCLUSION

We presented a hardware-aware MobileNetV2-lite system for BioDCASE-Tiny 2026 Task 3. The system keeps the official acoustic frontend, designs the student graph around ESP32-S3-friendly int8 operators, and uses a 19-model teacher pool to improve the compact student's class-discriminative supervision. The final int8 model improves validation accuracy and ROC-AUC over the official embedded baseline while substantially reducing model size, MACs, memory usage, and board inference time. These results indicate that matching the network graph to the target microcontroller runtime is as important as reducing parameter count for practical TinyML bird-sound recognition.

7. REFERENCES

[1] C. Walter, Y. Benhamadi, T. Seidel, G. Carmantini, and S. Kahl, "Biodcase-tiny 2026: A framework for bird species recognition on resource-constrained hardware," https://github.com/birdnet-team/BioDCASE-Tiny-2026, 2026, software, GitHub repository.

[2] G. Carmantini, Y. Benhamadi, M. Carreau, M. Kwak, I. Morandi, F. Förstner, P.-E. Hladik, M. Lagrange, P. Linhart, T. Petrusková, V. Lostanlen, and S. Kahl, "Bioacoustics on tiny hardware at the biodcase 2025 challenge," in *Proceedings of the 10th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2025)*, Barcelona, Spain, October 2025, pp. 80–84.

[3] Espressif Systems, "Esp-nn: Optimized neural network functions for espressif chips," https://github.com/espressif/esp-nn, 2026, software, GitHub repository.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[5] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, V. Massimino, M. Zhu, Y. Zhang, P. Anchuri, A. Howard, and M. Sandler, "Mobilenetv4: Universal models for the mobile ecosystem," in *Computer Vision – ECCV 2024*. Springer Nature Switzerland, 2025, pp. 78–96.

[6] S. Kahl and R. Martin, "Biodcase 2026 task 3: Bioacoustics for tiny hardware development set," https://doi.org/10.5281/zenodo.19453065, 2026.