CONVOLUTIONAL NEURAL NETWORK WITH KNOWLEDGE DISTILLATION FOR RESOURCE-CONSTRAINED BIOACOUSTICS

Technical Report

Naveen Dhar

High Tech High Mesa
San Diego, CA 92111, USA
naveendhar8030@gmail.com

ABSTRACT

The shift toward "tiny" machine learning enables the deployment of sophisticated audio classification models directly on power-efficient hardware, offering numerous benefits: immediate threat detection, lower latency for conservation insight, and vastly increased operational lifetimes in remote environments. However, these advantages come with constraints on memory, computational power, and energy consumption, necessitating unique approaches in both model architecture and feature extraction, in an environment where traditional deep learning methods negate practicality despite accuracy. A pipeline utilizing knowledge distillation and a custom Convolutional Neural Network (CNN) was developed for resource-constrained bioacoustics and evaluated on Task 3 of the 2025 BioDCASE challenge: detecting Yellowhammer bunting vocalizations in near and far-field environments using a lightweight model. The proposed network was built upon the BioDCASE baseline system, utilizing MobileNet convolution and depthwise convolution blocks, giving the name "SlimCNN". A larger "SlimCNN" was used as a teacher, and a smaller version was used as a student during knowledge distillation. Due to the nature of the task, ways to improve or maintain performance while maintaining or lowering architecture size, such as data augmentation, weight pruning, and adjustment of feature creation parameters, were given attention. The proposed "SlimCNN" achieved a 0.994 average precision on the provided validation dataset and a 13KB file size when quantized, demonstrating the potential for efficient machine learning.

***Index Terms*—** Bioacoustics, BioDCASE, tiny-ML, autonomous recording units, CNN, GRU, CRNN, knowledge distillation

1. INTRODUCTION

The deployment of machine learning models on resource-constrained hardware platforms represents a critical frontier in bioacoustic monitoring, where traditional deep learning approaches sacrifice practicality for accuracy. Autonomous recording units deployed in remote environments demand models that balance classification performance with severe constraints on memory footprint, computational complexity, and power consumption. This challenge becomes particularly acute in conservation applications where immediate threat detection and extended operational lifetimes are paramount. Task 3 of the 2025 BioDCASE challenge addresses this fundamental tension by requiring participants to develop lightweight models capable of accurate bird vocalization detection. This investigation presents a comprehensive model compression pipeline that integrates knowledge distillation, magnitude-based pruning, and quantization-aware training to achieve efficient avian call classification. The approach maintains robust performance across varying signal-to-noise conditions and recording distances through strategic architecture design, data augmentation, and training methodologies.

The competition utilized a curated dataset of Yellowhammer bird vocalizations, encompassing over two hours of two-second recordings sampled at 16kHz. The dataset comprised songs from multiple individual birds captured at varying recording distances, spanning 6.5 to 200 meters, which included a wide range from low to high Signal-to-Noise Ratio (SNR) recordings. Recordings were also obtained from different habitat types, specifically forest and grassland environments. The dataset incorporated negative samples, consisting of other bird species vocalizations along with background noise. Across the entire dataset, calls from twelve individual Yellowhammer birds were recorded and partitioned by individual. The training set contains recordings from eight individual birds, while the validation set includes recordings from two separate individuals. Two additional individuals are designated for the final evaluation phase.

2. METHODS

2.1. Preprocessing

The challenge required participants to follow a predefined preprocessing system that generated log-mel spectrograms from waveforms, but spectrogram parameter adjustment was allowed and encouraged. A window length of 1024 samples out of 16kHz*2sec or 48000 samples was chosen, with 50% overlap or a window stride of 512 samples. Frequencies were bounded between 1000Hz and 7500Hz, enabling representation of harmonic features with 64 Mel-frequency bins.

The parameters produced spectrograms of length 61 (time) and width 64 (frequency), and offered a tradeoff between temporal-frequency resolution and input size.

2.2. Architectures

The "SlimCNN" consisted of a convolutional section and a classification section. The same network structure was used by both the teacher and student models during knowledge distillation, the only difference being filter size. The general architecture starts with one initial MobileNet convolutional block, followed by four depthwise

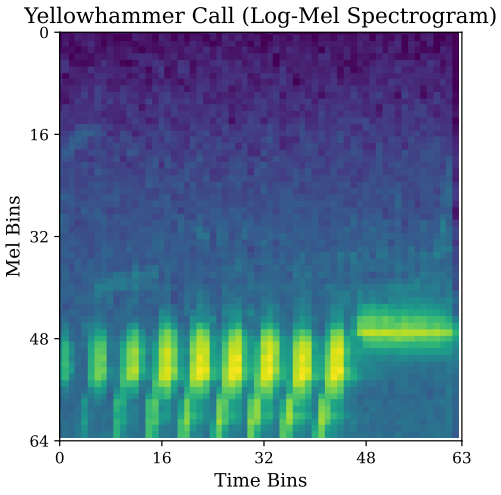

Figure 1: Example of Yellowhammer call displayed through a Log-Mel spectrogram using the parameters described in **??**.

separable convolutional blocks. These convolutional layers were followed by a two-dimensional global max pooling layer that maintained dimensions and a flattening layer. The flattened features were then fed to a two-unit fully-connected layer with softmax activation. Dropout was used after the last depthwise convolutional layer. For the teacher model, both the initial convolutional block and the first depthwise convolutional block had 32 filters, and the remaining depthwise convolutional blocks each had 64 filters.

2.3. Augmentation

To reduce false-negative and false-positive classifications and improve generalizability, augmentations were performed dynamically. The BioDCASE Task 3 baseline system was modified to enable dynamical augmentation in the training process by removing the caching of the training dataset. This also allowed dynamic reshuffling of the training data, ensuring samples were not only slightly different in content but also appeared in a different order each epoch.

SpecMixup, or simply the overlaying of a negative spectrogram on top of a positive spectrogram, was performed alongside the addition of Gaussian noise, creating more examples of far-field or low-SNR recordings. Because completely reducing the discerning features of already low-SNR Yellowhammer calls completely through the augmentation process was unwanted, the intensity of the augmentation was filtered based on a simple calculation for an SNR estimate. For each positive spectrogram in every training dataset batch, the flattened max and median were calculated. If the difference between the two was greater than the 70th percentile across all positive samples, background noise and SpecMixup augmentation bounds were modified, increasing in intensity. This produced low-SNR Yellowhammer recordings only from pools of relatively high-SNR Yellowhammer recordings. Each augmentation had a 30% chance of occurring, and neither negative samples nor validation dataset samples were augmented.

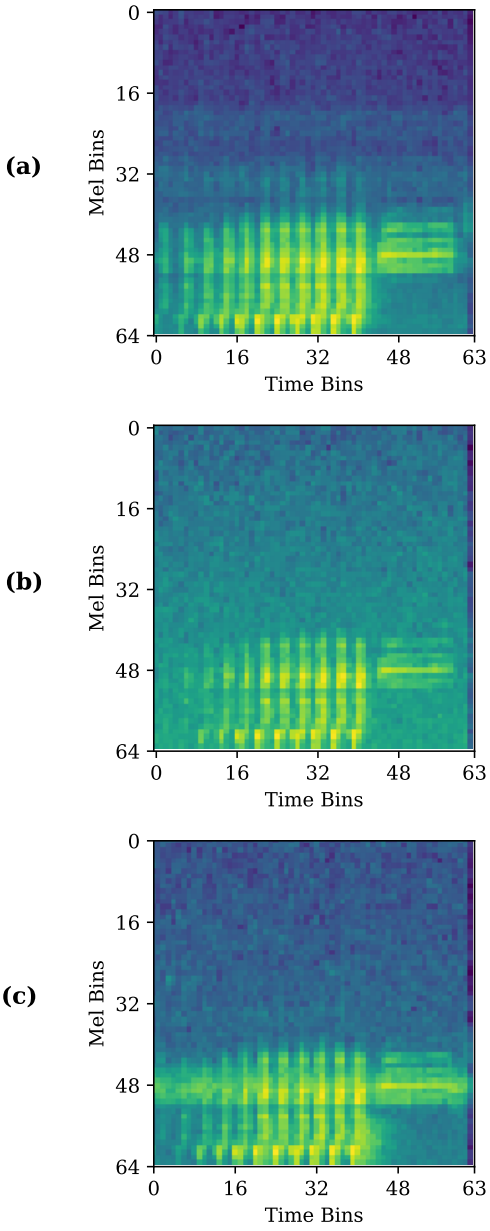

Figure 2: A display of the augmentations performed. The top spectrogram (a), has no augmentation. The middle spectrogram (b) has Gaussian noise augmentation. The bottom spectrogram (c) has overlay augmentation.

2.4. Knowledge Distillation and Training

Knowledge distillation was employed to dramatically reduce file size with minimal loss in performance, crucial for tasks such as real-time edge deployment for resource-constrained bioacoustics.

The teacher "SlimCNN" model was trained for 30 epochs using the Adam optimizer, binary cross-entropy loss, and a batch size of 32. The Keras model checkpoint callback was used to save the epoch with the best validation average precision. This model checkpoint served as the teacher model during knowledge distillation, transferring the learned representations to the student model.

The student model architecture was a slimmed version of the teacher "SlimCNN" architecture: only two depthwise convolutional layers instead of four and a 50% compression of the convolutional filters.

The distillation framework implemented a combined loss function:

$$L_{TOTAL} = \alpha \times L_{KD} + (1 - \alpha) \times L_{CE} \quad (1)$$

Where L_{KD} represents the Kullback-Leibler divergence between teacher and student soft predictions, L_{CE} denotes the standard cross-entropy loss, and α (set to 0.5) balances the contribution of each component. Temperature scaling softened the teacher's probability distributions and was used with a scaling parameter of 3.0.

During knowledge distillation training, cosine decay was used to schedule the learning rate. Training occurred for 25 epochs using the Adam optimizer, an initial learning rate of 0.001, and a batch size of 32. The same augmentation parameters used for training the teacher model were also applied during knowledge distillation training to ensure the student model would be robust to potential low-SNR recordings.

2.5. Pruning and Quantization

To further reduce model file size, the student model's weights were pruned following distillation.

Structured pruning using Tensorflow Model Optimization eliminated redundant network connections through magnitude-based weight reduction. The pruning schedule implemented polynomial sparsity increase from 0% to 35% over 8 epochs, or 896 training steps, selectively targeting convolutional and dense layers while preserving batch normalization and activation functions. Training during pruning utilized moderate data augmentation, with reduced probabilities ($p = 0.15$).

The final compression stage implemented Quantization-Aware Training (QAT) to support INT8 inference deployment for resource-constrained edge devices.

The QAT process annotated network layers with fake quantization operations, simulating reduced precision arithmetic during forward propagation while maintaining full precision gradients for backpropagation. Batch normalization layers received no quantization configurations to prevent numerical instability, while remaining layers underwent standard quantization annotation. QAT ran for six epochs with the Adam optimizer and a small learning rate of 0.0001. A minimal level of augmentation ($p = 0.05$) was chosen during QAT to stabilize convergence amid the noise introduced from the QAT process.

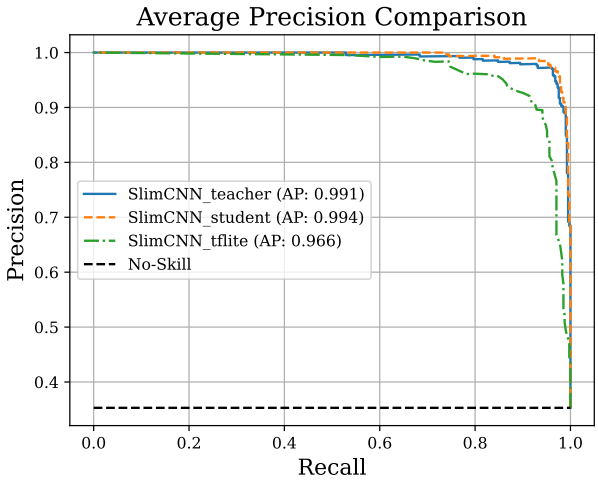

Figure 3: Precision-Recall curves for the proposed "SlimCNN" models on the validation dataset. "AP" denotes average precision. Knowledge distillation retained almost all original performance.

Model	File Size	Average Precision
"SlimCNN" Teacher	151KB	0.991
"SlimCNN" Student	69KB	0.994
"SlimCNN" TFLite	13KB	0.966

Table 1: Table illustrating model file size reduction alongside performance. Metrics were derived from the validation dataset, and sizes for the non-TFLite models were derived from the file size of a weights-only model.

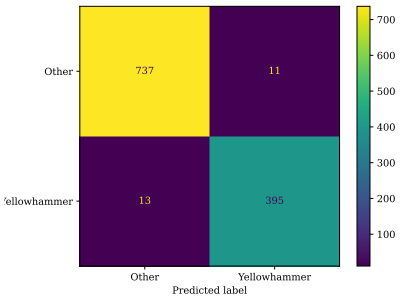

Figure 4: The confusion matrix on the validation dataset for the "SlimCNN" student model. Classification made using the threshold that produces the highest F1.

3. RESULTS

The proposed SlimCNN architecture demonstrated robust classification performance across the stages of the compression pipeline, illustrated in Fig. **??**. The teacher model achieved 0.991 average precision, with knowledge distillation, interestingly, improving performance to 0.994 average precision in the student model. This performance improvement, while uncommon in knowledge distillation, can possibly be attributed to the regularization effect of the distillation process, acting as a fine-tuning process. The confusion matrix (Figure 4) reveals an error rate of 2.5%, defined as:

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

The low misclassification rate demonstrates the architecture's discriminative capacity for Yellowhammer vocalizations, aided by dynamic augmentation. As seen in Table **??**, the final quantized model achieved 0.966 average precision at a file size of 13KB, representing a reduction of 91.4% from the teacher model size while retaining 97.5% of the teacher's performance. This indicates effective knowledge transfer and robust feature representation learning throughout the compression process.

4. CONCLUSION

This investigation demonstrates that systematic model compression techniques can achieve substantial reductions in computational requirements while preserving classification performance for bioacoustic applications. The proposed SlimCNN architecture, compressed through knowledge distillation and quantization-aware training, achieved an approximately 12-fold reduction in model size from 151KB to 13KB while retaining 97.5% of the original performance (0.966 vs 0.991 average precision). The sequential compression pipeline—employing progressive augmentation strategies from full intensity during distillation to minimal augmentation during quantization—proved effective in maintaining feature learning robustness throughout the optimization process. These results suggest significant potential for deploying sophisticated acoustic monitoring systems on ultra-low-power hardware platforms, enabling extended autonomous operation in remote conservation environments. The framework's success on Yellowhammer detection indicates broader applicability to diverse bioacoustic classification tasks, where the balance between model sophistication and deployment constraints remains a persistent challenge. Future work should investigate the scalability of this approach across multi-species detection scenarios and evaluate real-world deployment performance on embedded hardware platforms to validate the practical implications of these compression techniques for conservation monitoring applications.