EfficientNet-Based Multi-Label Audio Classification for BirdCLEF 2025

Jessie Zhang
University of California, San Diego
jiz199@ucsd.edu
A17669538

Caroline Zhang
University of California, San Diego
caz020@ucsd.edu
A69034335

Ting-Shiuan Lai
University of California, San Diego
t3lai@ucsd.edu
A69034403

Yixin Liu
University of California, San Diego
yil165@ucsd.edu
A16868457

Shuyu Wang
University of California, San Diego
shw043@ucsd.edu
A16357263

ABSTRACT

The BirdCLEF 2025+ [3] challenge addresses the task of identifying the presence of 206 animal species from real-world soundscape recordings collected in Colombia's El Silencio Reserve. Participants must detect species from noisy, overlapping, and often underrepresented vocalizations in one-minute audio clips. We propose a two-stage hybrid system that combines an EfficientNet-B0 image classifier trained on three-channel mel spectrograms with Sound Event Detection models designed to improve temporal and rare-event resolution. Our training pipeline includes extensive data preprocessing, spectro-temporal augmentations, and test-time ensemble smoothing. The final model achieved an AUC of 0.851 on the public leaderboard, demonstrating strong generalization in the multi-label acoustic classification setting.

1 INTRODUCTION

Problem Statement

In the BirdCLEF 2025+ [3] Kaggle competition, the goal is to develop a machine learning system capable of identifying animal species—spanning birds, amphibians, mammals, and insects—based on their acoustic signatures in real-world soundscape recordings. The primary data source comprises one-minute long audio clips recorded in the El Silencio Natural Reserve, a biodiversity hotspot in the Magdalena Valley of Colombia undergoing ecological restoration.

Formally, the task involves learning a function

$$f : \mathbb{R}^T \to [0, 1]^C$$

where $x \in \mathbb{R}^T$ represents an input audio waveform of length T (sampled at 32 kHz), and the model f outputs a probability vector over $C = 206$ species indicating the likelihood of each species' presence in a 5-second segment of the audio.

The competition presents several challenges:

- Multi-label classification: Multiple species can vocalize within a single audio segment.
- Data imbalance and rarity: Many species appear infrequently, with limited labeled samples.
- Real-world acoustic variability: Recordings contain substantial background noise, overlapping vocalizations, and habitat-specific distortions.
- Limited supervision: In addition to labeled training clips, participants are provided with large amounts of unlabeled soundscape data, enabling semi-supervised or self-supervised learning techniques.

The model's performance is evaluated on a hidden test set, using a custom scoring metric that rewards accurate detection across species. Success in this task supports conservation efforts by automating biodiversity monitoring and enabling high-resolution insights into ecological restoration progress.

2 BACKGROUND AND RELATED WORK

Acoustic Species Classification. Acoustic monitoring is an effective method for ecological research, allowing non-invasive and continuous detection of wildlife in natural environments. Deep learning models, especially those using convolutional neural networks (CNNs) on spectrogram inputs, have become standard for bioacoustic tasks [2, 7].

BirdCLEF Competitions. BirdCLEF, part of the LifeCLEF challenge series, has served as a benchmark for species classification from audio since 2014. It has evolved from simple bird identification to complex multi-label classification involving multiple taxa and noisy soundscapes. Past successful solutions have leveraged pretraining on large-scale datasets such as AudioSet [1].

Multi-label Learning and Data Imbalance. The BirdCLEF 2025+ task involves multi-label prediction, where several species may vocalize simultaneously. This setting introduces challenges such as label imbalance and the need to capture label dependencies. Techniques like focal loss [4] and semi-supervised learning (e.g., FixMatch) [6] have shown promise in improving generalization, especially when labeled data is limited.

Robustness in Natural Soundscapes. Field recordings contain noise from overlapping calls, wind, insects, and human activity. To address this, techniques such as spectral augmentation (e.g.,

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

SpecAugment) and noise-aware training are commonly used to improve model robustness [5].

2.1 Overview of the design

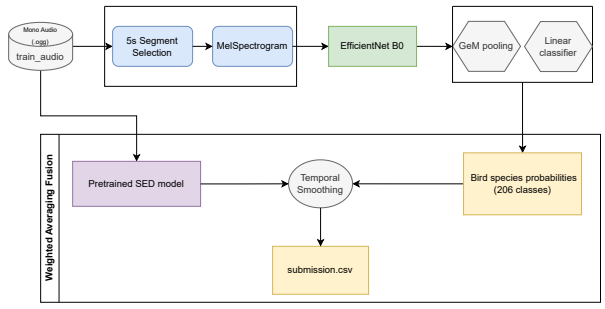

Figure 1: BirdCLEF model architecture

Our model, as shown in Figure 1, processes mono audio recordings by first applying `librosa` to extract log-Mel spectrograms, which are resized and saved as precomputed spectrograms for efficient training. These spectrograms undergo data augmentation (RandAugment, SpecAugment, mixup) and are fed into an EfficientNet-B0 backbone with GeM pooling and a linear classifier to produce multi-label bird species predictions. In parallel, a pretrained SED model processes the same spectrograms to generate complementary predictions. The outputs of both branches are fused via weighted averaging, followed by temporal smoothing across segments, to produce the final submission.

3 METHODOLOGY

3.1 Data Cleaning and Augmentation

Data Cleaning Data cleaning was conducted by addressing missing values within the audio signals. If any `NaN` values were present in an audio array, they were replaced with the mean of the valid (non-`NaN`) values to ensure numerical stability during subsequent processing steps. Each audio file was then loaded and truncated to the first six seconds. From this segment, a five-second window with the highest energy was selected, a heuristic aimed at isolating the most acoustically informative portion of the signal while minimizing the impact of silence or background noise.

Feature Extraction Following cleaning, each five-second audio segment was transformed into a mel spectrogram using Librosa's `melspectrogram` function. The resulting spectrogram was then converted to a decibel (dB) scale using `librosa.power_to_db(mel, ref=np.max)`, providing a perceptually meaningful representation of sound intensity relative to the signal's peak energy. To further standardize the inputs for model training, the spectrograms were normalized to a $[0, 1]$ range by subtracting the minimum and dividing by the dynamic range. Although this stage does not explicitly include augmentation techniques such as noise injection or time shifting, it establishes a consistent and information-rich input representation that facilitates effective downstream learning.

Data Augmentation Data augmentation was applied primarily to the 3-channel image representations of the mel spectrograms. A transformation pipeline was constructed using `torchvision.transforms`, which includes random augmentations via `RandAugment`. This module introduces a variety of image-level perturbations (e.g., rotations, brightness and contrast changes), followed by normalization to ImageNet statistics and the application of `RandomErasing`, which occludes random rectangular regions to simulate partial data loss. These augmentations were applied probabilistically during training to improve the model's robustness and generalization to novel inputs. Additionally, mixup augmentation was configured with `mixup_alpha = 0.5`, enabling the blending of input samples and their labels to further regularize the learning process.

Test-Time Augmentation (TTA) was employed during inference to enhance prediction robustness. This was implemented through the `apply_tta` function, which applies a set of deterministic and stochastic transformations to the input spectrograms. These include time shifting (via horizontal rolling), additive Gaussian noise to simulate ambient variability, and SpecAugment-style masking, where either a temporal or frequency segment of the spectrogram is zeroed out. One additional transformation flips the spectrogram along the time axis and amplifies its contrast to highlight feature boundaries. By averaging predictions across multiple augmented versions of the same input, the model achieves greater stability and resilience to minor input variations, thereby improving inference reliability on real-world acoustic data.

3.2 Three-Channel Image-Based EfficientNet-B0 Model

As an initial baseline, we built a complete classification pipeline using 3-channel log-mel spectrograms as input to an EfficientNet-B0 model. This setup allows us to evaluate how well standard image classification models, trained on natural images, perform when applied to bird audio data reformatted as image-like inputs. By using spectrograms as images and connecting them to pretrained image models, we can reuse proven visual architectures without having to design domain-specific acoustic structures.

The pipeline begins by dividing each long soundscape into 5-second clips, sampled at 32,000 Hz. For every clip, a log-mel spectrogram is computed using a 2048-point FFT window and a hop length of 512. The result contains 128 mel bands and spans frequencies from 20 Hz to 16,000 Hz. This representation captures key frequency components relevant to bird vocalizations. For efficiency, all spectrograms are precomputed and stored as `.npy` arrays.

Since models like EfficientNet-B0 expect three-channel RGB images, we convert the single-channel spectrograms into three channels by duplicating the matrix across the RGB dimensions. Each image is resized to 256×256 to meet the input size expected by the model. This conversion makes it straightforward to use visual backbones without modification.

The model uses EfficientNet-B0 as its backbone. We load pretrained weights from ImageNet using the `timm` library and replace the original classification head with a new fully connected layer, which outputs one logit per bird species. To improve feature aggregation before the final prediction layer, we apply Generalized Mean Pooling (GeM) instead of the standard average pooling. GeM is a flexible pooling function that raises feature values to a trainable

power p, averages them, and then takes the p-th root. When $p = 1$, it acts like average pooling; with larger p, it behaves closer to max pooling. The model learns p during training, which allows it to adaptively choose between spreading focus and highlighting strong activations. This is particularly useful in spectrograms, where informative regions may be small and localized.

Because bird calls often overlap, we treat the problem as multi-label classification. Each output logit corresponds to a species and is interpreted independently. The model is trained using method: `BCEWithLogitsLoss`. Optimization uses AdamW with a learning rate of 5×10^{-4}, weight decay of 1×10^{-5}, and cosine annealing to gradually reduce the learning rate to 1×10^{-6} across 15 epochs. Mixed-precision training is used to improve speed and reduce memory usage. We also apply typical image augmentations—random crops, horizontal flips, and mixup—to improve generalization and robustness.

All key configurations—such as FFT parameters, normalization, resizing, optimizer settings, and model structure—are managed in a centralized configuration system, ensuring consistency and reproducibility across experiments. This three-channel EfficientNet-B0 pipeline provides a straightforward but solid foundation for bird audio classification. It avoids unnecessary complexity while offering enough flexibility to support later additions like segmentation, ensemble learning, or integration with time-sensitive models such as SED. In early experiments, this baseline delivered stable results and served as a strong anchor for further development.

3.3 Sound Event Detection (SED) Model

Sound Event Detection (SED) is designed to identify and localize specific sound events within continuous audio streams, specifying not only which events occur but also when. Unlike standard audio classification models that produce a single prediction per audio clip, SED models work at a finer temporal resolution, making it possible to detect overlapping signals or transient vocalizations—characteristics often seen in ecological recordings such as bird soundscapes.

In our pipeline, we adopted SED models to complement the predictions generated by our baseline EfficientNet-B0 classifier. This choice stems from a common issue in bird call recognition: species with sparse annotations or brief call durations are frequently missed by standard models trained on clip-level labels. SED models, with their ability to learn time-localized patterns and capture low-level signal fluctuations, offer a way to address this limitation by providing additional temporal sensitivity.

Rather than training our own SED models from scratch, we chose to integrate three publicly available pretrained models released by the 5th place team in BirdCLEF 2025. These models are based on the ECA-NFNet architecture and were trained using a multi-stage self-distillation process. In this setup, predictions from an earlier model (the teacher) are used to generate pseudo-labels, which are then combined with original annotations to train a new model (the student). This process is repeated multiple times, gradually improving the model's ability to infer missing or secondary labels, particularly for underrepresented classes.

For inference, each of the pretrained models is instantiated as a `TimmSED` object and loaded in evaluation mode. Soundscape recordings are segmented into five-second clips and converted into mel spectrograms with 128 frequency bins. These spectrograms are then log-scaled and normalized to zero mean and unit variance. Since the SED models were trained on three-channel inputs (using ImageNet-pretrained weights), we replicate the single-channel spectrogram across all three channels to match the input format used during training.

The feature extraction process begins with the ECA-NFNet backbone, which processes the spectrograms and outputs frame-level representations. These features are aggregated across the time axis using both max pooling and average pooling. The pooled vectors are concatenated and passed through a fully connected layer with ReLU activation. A class-specific attention mechanism then computes weights across time steps, highlighting the temporal regions most relevant to each species, and uses these to produce final per-class logits.

After computing the logits, we apply a sigmoid function to obtain class probabilities. To improve the model's ability to detect rare or underrepresented classes, we apply a specific post-processing operation: `apply_power_to_low_ranked_cols`. This function targets predictions outside the top-K highest probability scores (we set $K = 30$) and raises them to a fixed exponent (we use exponent = 2). This transformation increases the influence of mid-confidence predictions—likely corresponding to rare species—without disrupting the predictions of dominant classes.

Once we obtain probability outputs from the three SED models, we average them to form an ensemble prediction. To enforce temporal consistency across clips, we apply a weighted moving average to the probability sequence of each species. For interior clips, we use a three-point smoothing kernel with weights of [0.2, 0.6, 0.2] applied to the previous, current, and next clip. For clips at the start or end of a recording, we use a two-point scheme with weights of [0.8, 0.2] or [0.2, 0.8]. This step reduces fluctuations and helps reflect the natural continuity of bird calls over time.

We chose to use these pretrained models rather than build our own due to both practical constraints and strategic considerations. First, high-quality SED training requires frame-level annotations, which are limited or inconsistent in BirdCLEF. Second, replicating the multi-stage distillation pipeline used by the original team would demand substantial computing resources and time. Given the proven performance of these released models, their integration into our system offered a reliable and efficient means to enhance classification accuracy without extensive retraining.

Incorporating SED outputs into our final inference ensemble enabled us to more effectively capture rare or short-duration calls that were often missed by the baseline model alone. This combination of temporally sensitive detection and global classification helped to improve overall system robustness, as reflected in our final leaderboard performance.

3.4 Inference

During the inference step, we load our trained EfficientNet-B0 model, instantiated as a BirdCLEFModel object with the GEM polling and linear classifier that we introduced above, and set it

Table 1: Summary of Model Evaluation Results

Phase	AUC
Training (Epoch 1)	0.67
Training (Epoch 10)	0.996
Validation Split	N/A
Public Leaderboard	0.851

Table 2: 4-Fold Cross-Validation Summary

Fold	Train AUC	Best Val AUC	Best Val Epoch
1	0.7688	0.7079	14
2	0.7543	0.7025	13
3	0.7611	0.7098	15
4	0.7495	0.7036	12

to evaluation mode. We load the testing dataset using the method as same method as we use in cutting them into 5-second chunks, data cleaning, augmentation, and also classify using three-channel mel spectrograms. After the dataset is ready, we use the predict-on-spectrogram function to predict them, which includes Test-Time Augmentation (TTA). This function converts each audio segment into an RGB-style spectrogram before putting it into our well-trained model, and we obtain the prediction and row-ids, which are also the species ids.

After we get the predictions from the EfficientNet-B0 model, we combine its prediction float number with the prediction number created by the SED model for each bird species with a particular weight. we have try many combinations and the best one is 0.7 EfficientNet-B0 model and 0.3 SED pre-trained model.

As soon as we combine them together, we will run the smooth-submission function before saving and creating a submission file. The smooth-submission function takes a per-segment prediction probability DataFrame and returns a new DataFrame in which each soundscape's sequence of probabilities has been smoothed over time. For each unique soundscapes group, the function extracts the block of probability values and constructs a smoothed version of the block. If there is more than one segment in the group, the first segment's probabilities are replaced by an 80 percent weighted average of its own value and 20 percent of the value of the next segment; the last segment is also replaced by 80 percent its value and 20 percent the value of the preceding segment. Every interior segment is replaced by a three-point weighted average: 20 percent of the previous segment, 60 percent of itself, and 20 percent of the subsequent segment.

Temporal smoothing improves prediction resilience by exploiting the continuity of bird vocalizations: smoothing the probabilities of each five-second interval against neighboring ones dampens noise-or artifact-caused transient peaks and fills low-confidence gaps during real vocalizations, generating a more continuous probability curve that more closely aligns with the continuum nature of real bird sounds and is consistent with assessment metrics charging non-consecutive detection penalties. x After all the above functions and processes, our submission is created and ready to be evaluated.

4 EVALUATION

For model training, we opted to use the full training set without a dedicated hold-out validation split, as our goal was to maximize the use of available data. The model was trained for 10 epochs, and the validation AUC reported during training showed an increasing trend from 0.67 in early epochs to 0.996 at epoch 10. However, this validation score is not a reliable indicator of generalization, as it was computed on the same data used for training and is therefore subject to overfitting. The actual model performance was evaluated on the public leaderboard of the BirdCLEF 2025 competition [3], where our best single model achieved a public AUC of 0.851. For our final submission, we used this best model trained on the full dataset, without ensembling, as we expected that using the full training data would offer better generalization than any partial fold-trained model in our case. The observed gap between internal AUC and leaderboard AUC was anticipated, given the lack of a proper validation split.

We also trained our model both on the full training set and using 4-Fold Stratified K-Fold cross-validation for internal evaluation. The full-data model was trained for 10 epochs and achieved a training AUC of 0.996, but this was not a reliable measure of generalization as shown in as shown in Table 1. Its public leaderboard AUC was 0.851.

Our 4-Fold cross-validation runs used 15 epochs per fold, as shown in Table 2. For Fold 1, we observed the validation AUC improving steadily from 0.5343 at epoch 1 to 0.7079 at epoch 14, with the final AUC being 0.7016 at epoch 15. Similar trends were observed in other folds, with early instability (low AUC) followed by consistent improvement across epochs. Each fold required approximately 4 hours to train, making full ensembling computationally infeasible within the competition timeline. We therefore submitted the single full-data model for the final competition entry.

5 LIMITATION

One aspect we explored was the effect of human voice removal on model performance. It is well known that segments with human voice can bias model learning and degrade generalization, especially for rare species. In BirdCLEF 2025, several top teams implemented careful voice removal pipelines. For example, one team used Silero VAD to detect audio files with human voices, followed by manual verification with a Streamlit tool to precisely remove voice segments. For underrepresented classes (with fewer than 30 samples), they manually selected segments containing bird calls to preserve valuable data. Their final strategy used the first 60 seconds for cleaned files and the first 30 seconds for others, along with dataset balancing through duplication of low-sample classes.

In our case, we also attempted to apply Silero VAD to automatically filter out segments containing human voice. However, our fully automatic approach—without manual verification or manual segment selection—did not yield improvements; in fact, it slightly lowered our AUC scores. We hypothesize that noisy VAD predictions and aggressive removal of certain segments may have inadvertently discarded useful bird call information. Additionally, since we trained only on the official BirdCLEF 2025 dataset (without external data), our ability to rebalance the dataset post-removal was limited.

Moving forward, we believe that combining VAD with selective manual verification and segment-level cleaning, as demonstrated by other teams, could potentially yield better results in future iterations of our pipeline.

Another potential direction is to incorporate a self-training pipeline to further improve rare species detection and leverage unlabeled data. Specifically, we plan to adopt a Multi-Iterative Noisy Student approach, using our existing model ensemble to generate high-quality pseudo-labels on unlabeled soundscapes. Pseudo-labels will be smoothed and power-transformed to mitigate label noise, and weighted sampling will prioritize informative soundscape segments. The training process will employ MixUp between labeled and pseudo-labeled data to enhance generalization. We also intend to train a dedicated model for Amphibia and Insecta species using extended Xeno-Canto data and optimized Mel-spectrogram parameters to better capture their long, repetitive calls. Finally, inference will be enhanced by averaging overlapping framewise predictions with temporal smoothing and delta shift test-time augmentation, providing more robust and accurate predictions across all species.

6 CONCLUSION

In this project, we tackled the BirdCLEF 2025+ challenge by designing a hybrid classification pipeline that integrates CNN-based image models with SED-based temporal detectors. By converting audio into three-channel mel spectrograms and applying a variety of augmentations and smoothing strategies, we built a model that is both robust and scalable. Ensembling EfficientNet-B0 with three SED models allowed us to balance high-confidence predictions with improved rare species detection. Our final system achieved a public leaderboard AUC of 0.851, validating the effectiveness of our approach.

REFERENCES

[1] Jort F et al. Gemmeke. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP*.

[2] Shawn et al. Hershey. 2017. CNN architectures for large-scale audio classification. In *ICASSP*.

[3] Holger Klinck, Juan Sebastián Cañas, Maggie Demkin, Sohier Dane, Stefan Kahl, and Tom Denton. 2025. BirdCLEF+ 2025. https://kaggle.com/competitions/birdclef-2025. (2025). Kaggle.

[4] Tsung-Yi et al. Lin. 2017. Focal loss for dense object detection. In *ICCV*.

[5] Daniel S et al. Park. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*.

[6] Kihyuk et al. Sohn. 2020. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.

[7] Dan et al. Stowell. 2019. Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge. *Methods in Ecology and Evolution* (2019).

A ACKNOWLEDGMENT

We are sincerely grateful to Prof. Berk Ustun at the University of California, San Diego, for his continuous support, encouragement, and insightful guidance throughout this class. We also deeply appreciate the support of TA Ryan Hammonds, who assisted us throughout the quarter and patiently answered our questions. Many of the ideas presented in this report were inspired by discussions on the Kaggle BirdCLEF 2025 Discussion Board, and we would like to express our gratitude to the entire community for their valuable contributions.

B APPENDIX

Throughout this project, we explored various configurations of preprocessing, spectrogram parameters, and input selection strategies to optimize our model performance. Table **??** summarizes the key combinations we tested and their corresponding public leaderboard AUC scores. All experiments used the same core CNN architecture but varied in data preparation and augmentation techniques.

In addition to baseline experiments, we also implemented a human voice removal pipeline using VAD-based filtering with Silero-VAD.[1] For segments containing human voice, we selected alternative audio chunks based on highest RMS energy while avoiding overlap with voice regions. The results from this approach are also included below.

Experiment	Public AUC
Baseline	0.773
Baseline + Larger Hop	0.776
Baseline + Voice Removal	0.781
Best 5s Window	0.779
Best 5s + Voice Removal	0.784
Center Crop	0.766
Center Crop + Voice Removal	0.777
Optimized Mel	0.796
Optimized Mel + Voice Removal	0.806

Observations.

- Applying human voice removal consistently improved performance across multiple configurations (typically by +0.005 to +0.010 AUC).
- Increasing input duration to 20 seconds with larger hop size and more mel bands yielded substantial gains (+0.02 AUC over baseline), especially for species with long repetitive calls (e.g., Amphibia and Insecta groups).
- Center cropping without voice removal underperformed compared to best-5s selection strategies.
- The combination of 20s chunks + optimized Mel parameters + voice removal yielded our best result (0.806 AUC).

[1] https://github.com/snakers4/silero-vad