A MULTI-RESOLUTION HTS-AT TRANSFORMER WITH FEATURE-PYRAMID FUSION FOR ANTARCTIC BLUE AND FIN WHALE CALL DETECTION

Technical Report

Matija Marolt and Eva Boneš

University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, 1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

ABSTRACT

We describe our submission to BioDCASE 2026 Task 2 (Antarctic blue and fin whale call detection). Our model is based on the HTS-AT transformer, which takes a multiresolution input and produces per-frame multi-label classification of whale calls into three groups (plus background), as defined by the challenge. After preprocessing designed to compress the dynamic range of recordings, audio is processed by a learnable LEAF front-end. The model is a three-stream multi-resolution HTS-AT Swin transformer with a feature-pyramid fusion head, trained with an asymmetric multi-label loss on background-subsampled per-event clips. At inference we decode per-frame probabilities, apply a decision threshold and a per-class median filter, and convert detections into timed events. The model was trained exclusively on the data provided by the challenge and reaches a frame-level macro average precision of 0.69 on the test split. Three slightly different versions were selected for submission, and we also include results of our last-year's submission for comparison.

***Index Terms*—** whale call detection, transformer, multi-resolution, HTS-AT, BioDCASE

1. INTRODUCTION

BioDCASE 2026 Task 2 requires the temporal localisation of Antarctic blue (*Balaenoptera musculus*) and fin (*Balaenoptera physalus*) whale vocalisations in long passive-acoustic recordings. The seven annotated call sub-types are grouped into three acoustically coherent foreground classes: Bm-A/Bm-B/Bm-Z tonal calls (`bma`), Bm-D and Bp-D downsweeps (`bmd`), and Bp-20/Bp-20plus pulses (`bp20`). We approach the task as a per-frame multi-label classification problem, as different call types may co-occur in time.

Our approach consists of: (i) a preprocessing stage that filters out low frequency noise and reduces dynamic range; (ii) a multi-resolution HTS-AT transformer model with feature-pyramid fusion that produces dense per-frame predictions; and (iii) a simple post-processing stage (thresholding and per-class median filtering) that turns the frame probabilities into events. Figure 1 summarises the complete pipeline.

2. DATA PREPROCESSING

The recordings are sampled at 250 Hz (a small number of the 2026 evaluation files are at 1 kHz and are first downsampled to 250 Hz).

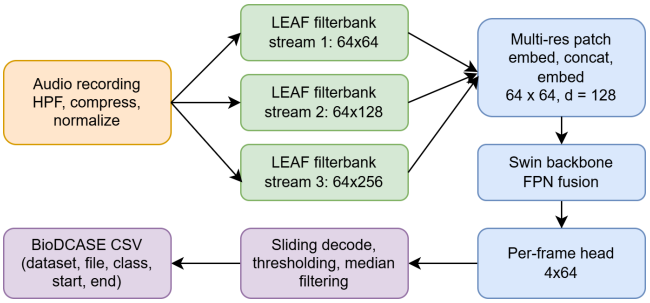

Figure 1: End-to-end pipeline: audio preprocessing (orange), the three-stream multi-resolution LEAF + HTS-AT FPN model (green/blue), and inference post-processing (purple).

Each file is processed by the chain shown in the top of Fig. 1:

1. DC offset removal (mean subtraction).
2. A high-pass filter is applied at 9 Hz to suppress low-frequency noise.
3. Each recording is dynamic-range compressed with the threshold set per file to the 99.9-th percentile of the original signal (ratio 8, 64 ms attack/release), so that only the loudest 0.1% of samples are compressed. This reduces large peaks that sometimes occur in the signals.
4. Loudness is normalised to a target of -30 dB RMS, capped so that the peak stays below -1 dB. This makes all recordings comparably loud.

3. MODEL

The model is a three-stream multi-resolution variant of the HTS-AT audio spectrogram transformer [1] with a feature-pyramid (FPN) fusion head. It has ≈ 54 M parameters and is initialised from publicly available AudioSet HTS-AT weights (loaded non-strictly, so the patch-embedding and head that differ from the pretrained model are trained from scratch).

3.1. Learnable front-end and multi-resolution streams

The acoustic front-end is LEAF [2], a learnable Gabor filterbank with 64 filters followed by per-channel PCEN compression. Three parallel front-end streams are run at progressively finer

time resolution: window/hop of $1.024/0.256$, $0.512/0.128$ and $0.256/0.064\,\mathrm{s}$. With input window size of $16.3\,\mathrm{s}$ this yields feature maps of size 64×64, 64×128 and 64×256 (frequency $\times$ time). The coarse stream captures the long tonal Bm calls while the finer stream better localise the short Bp pulses and downsweeps.

3.2. Multi-resolution patch embedding

Each stream is patch-embedded (4×4 patches) with per-resolution and per-modality stems, and projected onto a common 64×64 token ($d = 128$) grid. The three streams are fused by concatenation along the channel dimension, and a learnable per-scale embedding is added to disambiguate the streams, followed by a linear projection to fuse the three representations, producing a single token sequence with embedding dimension $d = 128$.

3.3. Swin backbone and FPN head

The fused tokens are processed by the HTS-AT Swin transformer backbone with depths $[2, 2, 6, 2]$, attention heads $[4, 4, 16, 32]$, and a window size of 8. The per-stage feature maps are merged by a feature pyramid network (256 channels) and upsampled by a classification head into dense per-frame logits. With a per-frame stride of $256\,\mathrm{ms}$ over the $16.3\,\mathrm{s}$ window, this yields 64 frames; the head emits four logits per frame (three foreground classes plus background). A sigmoid activation is used for multi-label output.

4. BATCHES AND TRAINING

4.1. Batch formation

Training examples are clips ($16.3\,\mathrm{s}$) randomly sampled within annotated events and background in each batch. Because background frames vastly outnumber call frames, each epoch draws a fresh random subset of background samples, so that foreground constitutes $\approx 25\%$ of the clips. In addition, inverse-frequency class rebalancing of foreground classes may be applied.

On-the-fly augmentations include waveform mixup and cutmix, SpecAugment, additive Gaussian and background noise clips, bandpass/band-stop/high-pass/low-pass filtering, gain changes and parametric equalisation.

4.2. Training

Training minimises an asymmetric multi-label loss [3] (ASL) on the per-frame logits. ASL applies a focusing exponent that differs between positives and negatives and shifts the negative probability to down-weight the easy, abundant background frames, which are the dominant source of imbalance in this task.

The network is optimised with AdamW (learning rate 10^{-4}, betas $(0.9, 0.98)$, weight decay 10^{-4}) under a Noam hold-anneal schedule (warm-up 5%, hold 20%, decay rate 0.5), for 50 epochs.

5. INFERENCE AND POST-PROCESSING

At inference the model is applied with hop size of $0.512\,\mathrm{s}$; overlapping windows are combined by averaging their post-sigmoid probabilities. The per-frame probabilities are then post-processed (bottom of Fig. 1):

1. **Median filtering.** A per-class temporal median filter is applied with kernel sizes $[9, 3, 3]$ frames for `(bma, bmd, bp20)`: the long tonal Bm class is smoothed more aggressively, the short Bp/Bm-D classes only lightly.
2. **Thresholding.** Each foreground track is binarised at a probability threshold.
3. **Event extraction.** Contiguous runs of active frames per class become events.

Table 1: Development-set results.

Metric	Value
Macro AP	0.69
Macro F_1	0.65
Micro F_1	0.53
Instance F_1	0.60
IoU@0.3	0.53

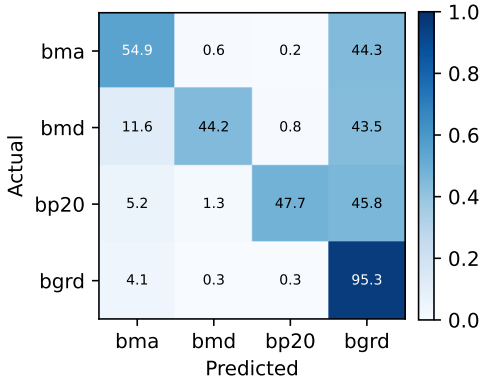

Figure 2: Row-normalised per-frame confusion matrix (%). The three call types are well separated from one another; the dominant error is confusion between foreground calls and background.

6. RESULTS

Table 1 presents metrics calculated by inferring the model on the entire files included in the official development set validation split, in the same manner as the model is run on the held-out dataset for submission (only without median filtering). The validation set was not used for early stopping; however, model selection was optimised based on validation set metrics, so the numbers are not entirely fair.

Per-frame AP, Micro and Macro F_1 are shown, as well as instance-based F_1 and IOU metrics. The confusion matrix (Fig. 2) shows that the three foreground call types are rarely confused with each other; the principal error lies in distinguishing the boundary between foreground calls and background, with $44 - 46\%$ of foreground frames falling into the background class.

7. SUBMISSION

Four model outputs were submitted for the challenge: three based on models described in this report and our submission from last year [4]. All new models were trained for 50 epochs on the entire 2026 development set, while last year's submission was simply reevaluated on the new 2026 evaluation set. The submissions are as follows:

1. Model trained without class rebalancing and inferred with threshold 0.6, which should result in better precision;
2. The same model as in (1), but inferred with threshold 0.5, which should result in better recall;
3. Model trained with class rebalancing and inferred with threshold 0.6. Class rebalancing should yield better performance if the evaluation split has a different class distribution than the development dataset;
4. Last year's submission, evaluated in the multi-label regime with threshold 0.5.

8. REFERENCES

[1] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proc. IEEE ICASSP*, 2022, pp. 646–650.

[2] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, "LEAF: A learnable frontend for audio classification," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

[3] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 82–91.

[4] M. Marolt and E. Boneš, "Multi resolution feature fusion for supervised detection of strongly-labelled whale calls," in *BioDCASE 2025 challenge*, 2025.