CLASS-AWARE HETEROGENEOUS SPECTROGRAM DETECTORS FOR BIODCASE 2026 WHALE CALL EVENT DETECTION

Technical Report

Yongyi Deng[1,†], *Yuxin Lei*[1,†], *Kai Pi*[1,†], *Yuzhu Wang*[2], *Zhuo Liu*[3]
Wei Liu[1], *Xu Shen*[4], *Xueying Tan*[1], *Gongping Huang*[1*]

[1] Wuhan University, Wuhan, China
[2] Tampere University, Tampere, Finland
[3] Northwestern Polytechnical University, Xi'an, China
[4] Waseda University, Japan
[†] Equal contribution.

ABSTRACT

This technical report describes our submission to BioDCASE 2026 Task 2 on whale-call detection from underwater acoustic recordings. We formulate event-level whale-call detection as a spectrogram-based object detection problem, where each call event is represented by a bounding box in the time-frequency domain and then mapped back to temporal boundaries in the original audio recording. The proposed system combines two heterogeneous detection streams with class-aware branch selection. A YOLO ensemble branch is used to generate candidates for the *d* and *bp* categories, while a cascade region-based Boundary-Aware Feature Fusion Network branch is selected for the *bmabz* category according to validation performance. The Boundary-Aware Feature Fusion Network branch is implemented with a DetectoRS ResNet-50 backbone, a Recursive Feature Pyramid neck, and a three-stage Cascade region-of-interest head, and is trained using Microsoft Common Objects in Context-style annotations converted from the original YOLO-format spectrogram labels. Predictions from the selected branches are converted into event-level candidates and refined by a unified post-processing pipeline, including class-aware filtering, temporal boundary adjustment, temporal non-maximum suppression, low-threshold candidate recovery, and deterministic output ordering. On the validation split, the selected Boundary-Aware Feature Fusion Network branch achieves an F1 score of 0.6616 for the retained *bmabz* candidates, while the YOLO ensemble branch achieves a micro F1 score of 0.6339 for the selected candidate set. These results indicate that heterogeneous detector branches and validation-based class-aware selection provide complementary behavior for spectrogram-based whale-call event detection.

***Index Terms*—** BioDCASE 2026, whale-call detection, passive acoustic monitoring, spectrogram object detection, YOLO ensemble, BAFNet, Cascade R-CNN, event-level post-processing

1. INTRODUCTION

BioDCASE 2026 Task 2 focuses on event-level whale-call detection from underwater acoustic recordings [2]. The goal is to predict both the call category and the temporal boundaries of each event. In this work, we formulate the task as a spectrogram-based object detection problem, where whale-call events are represented as bounding boxes in the time-frequency domain and then mapped back to temporal intervals in the original audio recordings.

This formulation allows mature object detection models to be adapted to whale-call event localization. However, the task remains challenging because whale calls may be weak, noisy, and variable in duration, and different call types exhibit different acoustic characteristics [3, 2]. In addition, the official event-level evaluation follows a one-to-one matching protocol, where duplicate predictions around the same ground-truth event are counted as false positives.

Our submitted system combines two heterogeneous spectrogram-based detection branches. The YOLO ensemble branch is used to generate candidates for the *d* and *bp* categories, while the Boundary-Aware Feature Fusion Network (BAFNet) branch is selected for the *bmabz* category according to validation performance. The selected candidates are further refined by a unified event-level post-processing pipeline, including class-aware filtering, temporal boundary adjustment, temporal non-maximum suppression (NMS), low-threshold candidate recovery, and deterministic output ordering.

2. PROPOSED SYSTEM

The proposed system follows a class-aware two-branch detection framework, as illustrated in Fig. 1. We first convert underwater audio recordings into spectrogram images and then apply two heterogeneous spectrogram-based object detection branches. The final prediction is not obtained by simply averaging all detector outputs. Instead, we select different detector branches for different whale-call categories based on their validation behavior. Specifically, the BAFNet branch is used for the *bmabz* category, while the YOLO ensemble branch is used for the *d* and *bp* categories. The selected candidate events are then converted into a unified event-level CSV format and refined by a common post-processing pipeline.

2.1. Submitted System

This report describes the submitted system `Deng_WHU_task2_2`. The system is named *Class-aware Heterogeneous Spectrogram Detector* and is abbreviated as *CHSDet_mix4*.

*Corresponding author.

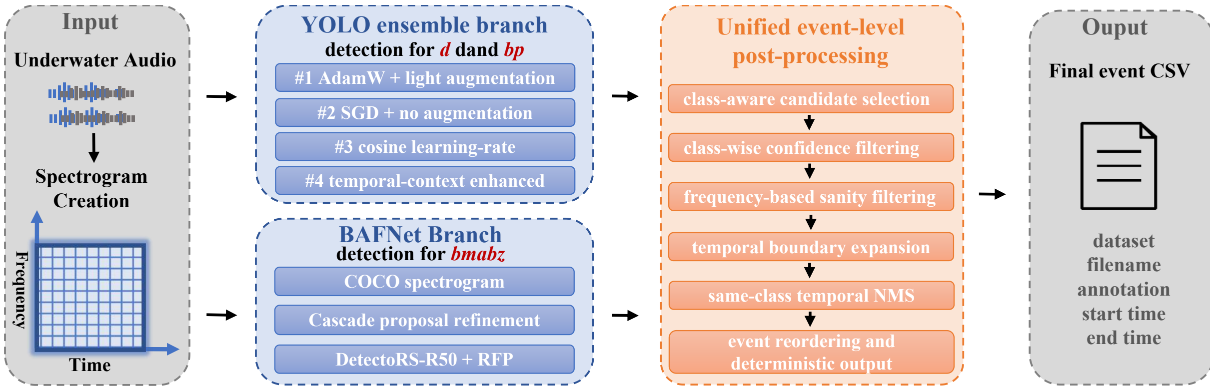

Figure 1: Overview of the proposed whale-call detection syste m. The system uses class-aware detector selection, where BAFNet is used for *bmabz* and the YOLO ensemble branch is used for *d* and *bp*, followed by unified event-level post-processing.

The metadata file records the submission label, system name, author information, acoustic representation, model configuration, inference strategy, external pretrained resources, and validation results. The submitted prediction file contains event-level predictions with the required fields: dataset, filename, annotation label, start time, and end time.

Table 1: Submitted system information.

Item	Description
Submission label	`Deng_WHU_task2_2`
Prediction file	`Deng_WHU_task2_2.csv`
Metadata file	`Deng_WHU_task2_2.meta.yaml`
System name	*Class-aware Heterogeneous Spectrogram Detector*
Abbreviation	*CHSDet_mix4*

No external audio data are used in this submission. The external resources listed in the metadata are limited to pretrained visual models or object-detection pretraining resources, including pretrained YOLO11s weights, COCO-related object-detector pretraining resources, and ImageNet-pretrained ResNet-50 weights from Torchvision. No evaluation-set annotations are used for model training, checkpoint selection, or post-processing parameter tuning.

2.2. Spectrogram-based Event Detection

We formulate whale-call detection as an object detection problem on spectrogram images [4]. For each audio recording $x[n]$, a time-frequency representation is generated using short-time Fourier transform (STFT):

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mH]w[n]\exp\left(-j\frac{2\pi kn}{N}\right), \quad (1)$$

where m and k denote the time-frame and frequency-bin indices, respectively, $w[n]$ is the analysis window, N is the FFT size, and H is the hop size. The spectrogram image is computed from the magnitude or log-magnitude of the STFT:

$$S(m,k) = \log\left(|X(m,k)| + \epsilon\right), \quad (2)$$

where ϵ is a small constant for numerical stability.

Whale-call events are then represented as bounding boxes on the spectrogram. For a predicted bounding box

$$b = (x_1, y_1, x_2, y_2, c, s), \quad (3)$$

x_1 and x_2 denote the left and right boundaries on the time axis, y_1 and y_2 denote the lower and upper boundaries on the frequency axis, c is the predicted class label, and s is the detection confidence. The class label c represents the whale-call category.

After detection, the horizontal coordinates of each predicted bounding box are converted back to the start and end times of the corresponding event. Assume that the spectrogram window has a duration of T seconds and an image width of W pixels. For the i-th spectrogram window with start time offset o_i, the predicted event boundaries are computed as

$$\hat{t}*\text{start} = o_i + \frac{x_1}{W}T, \quad (4)$$

$$\hat{t}*\text{end} = o_i + \frac{x_2}{W}T. \quad (5)$$

The final prediction is represented in the event-level CSV format required by the task, including the dataset name, filename, annotation label, start time, and end time.

2.3. YOLO Ensemble Branch

The YOLO ensemble branch is used to generate candidate events for the *d* and *bp* categories. It consists of four YOLO11s-based detectors with different training configurations, including three standard YOLO11s variants and one YOLO11s variant equipped with a Frequency-aware Temporal Context Module (FTCM). The first three detectors share the same YOLO11s architecture but differ in training strategies, such as optimization settings, learning-rate scheduling, training duration, and data augmentation. These variants are included to increase the diversity of the ensemble and improve the robustness of candidate event generation.

FTCM is inserted into the three output scales of the YOLO11s neck/head, corresponding to the P3, P4, and P5 feature levels. Given an input feature map of shape $[B, C, H, W]$, FTCM treats the width dimension W as the temporal axis and reshapes the feature map into a temporal sequence, where each time step contains the corresponding channel-frequency representation. A bidirectional gated recurrent unit (GRU) is then used to model temporal context along the spectrogram time axis. The resulting temporal feature is projected back to the original feature shape and combined with the input feature map through a learnable residual coefficient. This module provides additional time-frequency-aware temporal context modeling for spectrogram-based object detection.

The outputs of the four YOLO detectors are fused using same-class two-dimensional box fusion. For each class, overlapping bounding boxes from different detectors are grouped.For each class, overlapping bounding boxes from different detectors are grouped with an intersection-over-union (IoU) threshold of 0.5. The fused box coordinates are computed by confidence-weighted averaging, while the final confidence score is set to the maximum score within the cluster. In the final class-aware system, the YOLO ensemble branch is selected for the *d* and *bp* categories.

2.4. BAFNet Branch

The BAFNet branch forms the second detection stream of the proposed system [5]. While the YOLO ensemble branch is responsible for generating candidates for the *d* and *bp* categories, the BAFNet branch is used to detect *bmabz* events. Both branches operate on the same spectrogram images produced from the underwater recordings, but they follow different detection paradigms. This class-aware detector assignment is determined according to validation event-level performance, where BAFNet is selected for the *bmabz* category among the candidate branches.

The BAFNet branch is trained as a spectrogram-based object detector. The original YOLO-format bounding-box annotations are first converted into Microsoft Common Objects in Context (COCO)-style annotations [9] so that the data can be used in an MMDetection-style training pipeline [8]. The three merged whale-call categories, *bmabz*, *d*, and *bp*, are retained during training, although only the *bmabz* predictions from this branch are selected in the final class-aware system. This design keeps the detector trained under the same label space as the other branches while allowing branch selection to be performed at the event level.

The detector is implemented as a cascade region-based model, denoted as `CascadeRCNN_BAF`. It uses a DetectoRS ResNet-50 backbone, a Recursive Feature Pyramid (RFP) neck [6], and a three-stage Cascade RoI head [7]. Compared with the one-stage YOLO detectors, this cascade formulation refines object proposals through multiple stages and therefore introduces a complementary localization mechanism for spectrogram events. The detection heads are adapted to the three merged whale-call categories.

At inference time, BAFNet outputs class-specific bounding boxes and confidence scores for each spectrogram window. For each retained *bmabz* candidate, the horizontal box coordinates are mapped back to the temporal axis of the original audio recording, producing an event start time and end time. The epoch-6 checkpoint is used in the final system. Its bounding-box detections are converted into temporal events on the validation split, and the associated post-processing configuration for the retained BAFNet candidates is selected using the corrected one-to-one matching protocol. The resulting *bmabz* candidates are sent to the unified event-level post-processing module, where they are processed together with the *d* and *bp* candidates generated by the YOLO ensemble branch to produce the final event CSV.

Table 2: Class-dependent post-processing parameters used in the final system.

Class	Confidence	Frequency prior	NMS IoU
bmabz	0.30	10–45 Hz	0.15
d	0.22	15–120 Hz	0.30
bp	0.15	10–120 Hz	0.40

2.5. Unified Event-level Post-processing

The outputs of the YOLO ensemble branch and the BAFNet branch are first converted from spectrogram bounding boxes into event-level predictions:

$$p_i = (d_i, f_i, c_i, \hat{t} * \text{start}, \hat{t} * \text{end}, \hat{s} * i), \tag{6}$$

where d_i denotes the dataset name, f_i denotes the audio filename, c_i denotes the predicted category, $\hat{t}*\mathrm{start}$ and $\hat{t}*\mathrm{end}$ denote the event boundaries, and $\hat{s}_i$ is the confidence score. After branch-specific inference, all selected candidates are refined by a unified event-level post-processing pipeline, including class-aware candidate selection, confidence filtering, frequency filtering, temporal boundary expansion, temporal NMS [10], low-threshold candidate recovery, and deterministic event reordering. According to validation performance, the BAFNet branch is used for the *bmabz* category, while the YOLO ensemble branch is used for the *d* and *bp* categories.

The class-dependent post-processing parameters used in the final system are listed in Table 2. Before temporal NMS, the predicted start and end times are expanded by 0.5 s on both sides and clipped to the valid duration of the corresponding audio file.

Same-class temporal NMS is applied independently within each (d_i, f_i, c_i) group. Within each group, predictions are sorted by confidence, and lower-confidence events with excessive temporal overlap are suppressed. For two predictions p_a and p_b, temporal IoU is computed as

$$\text{IoU} * t(a,b) = \frac{I(a,b)}{U(a,b)}, \tag{7}$$

where $I(a,b)$ and $U(a,b)$ denote the temporal intersection and union of the two events, respectively. If $\text{IoU} * t(a,b)$ is larger than the class-dependent NMS threshold, the lower-confidence prediction is removed. Low-threshold predictions are used as a recall-compensation pool, where uncovered candidates are added as supplementary events and overlapping candidates are resolved by retaining the higher-confidence event. Finally, events within the same dataset, filename, and annotation group are reordered in descending order of confidence before generating the official CSV file. All post-processing parameters are selected on the validation split and fixed for final inference.

3. EXPERIMENTAL SETUP

3.1. Dataset

The experiments are conducted on the BioDCASE 2026 Task 2 whale-call detection dataset, which is derived from annotated Antarctic blue and fin whale recordings used for detector development and benchmark evaluation [1, 2]. Following the official evaluation setting, the original call annotations are merged into three categories: *bmabz*, *d*, and *bp*. Specifically, *bma*, *bmb*, and *bmz* are merged into *bmabz*; *bmd* and *bpd* are merged into *d*; and *bp20*

Table 3: Training configurations of the YOLO ensemble branch.

Model	Architecture	Optimizer / LR	Training duration	Augmentation
Quality YOLO11s	YOLO11s	AdamW, lr=0.001, cosine LR	about 40 epochs	light augmentation
Baseline YOLO11s	YOLO11s	SGD, lr=0.001	20 epochs	minimal augmentation
Cosine baseline	YOLO11s	SGD, lr=0.0003, cosine LR	8 epochs	minimal augmentation
FTCM YOLO11s	YOLO11s + FTCM	AdamW, lr=0.0001, cosine LR	about 40 epochs	spectrogram-friendly augmentation

and *bp20plus* are merged into *bp*. All detector branches and post-processing modules use this three-class setting.

After spectrogram generation and label conversion, the training split contains 115,017 spectrogram images and 113,154 bounding boxes, while the validation split contains 43,684 spectrogram images and 35,170 bounding boxes. The validation split is used for model selection and post-processing parameter selection. The evaluation set is only used for final inference and submission generation.

3.2. Training details

The YOLO ensemble branch consists of four YOLO11s-based detectors. All YOLO models are trained with the same three-class label setting and use an input size of 640 and a batch size of 32. They are initialized from pretrained weights and trained with different optimization schedules and augmentation settings to increase ensemble diversity. The detailed YOLO training configurations are listed in Table 3.

The quality-enhanced YOLO11s model uses light spectrogram augmentation, including moderate color perturbation, translation, scaling, horizontal flipping, and mosaic augmentation. The two standard YOLO11s baselines use minimal augmentation and provide stable reference detectors with different training schedules. The FTCM-enhanced YOLO11s model uses spectrogram-friendly augmentation and does not use rotation or horizontal flipping, since the time and frequency axes of spectrograms have explicit physical meanings.

The BAFNet branch is trained independently from the YOLO ensemble using COCO-style annotations converted from the YOLO-format spectrogram labels. The selected detector is `CascadeRCNN_BAF`, consisting of a DetectoRS ResNet-50 backbone, an RFP neck, and a three-stage Cascade RoI head. All detection heads are configured for the three merged call categories, *bmabz*, *d*, and *bp*. Input spectrogram images are resized with a target scale of 512×384 while preserving the aspect ratio, followed by normalization and stride-compatible padding.

The backbone is initialized with ImageNet-pretrained ResNet-50 weights from Torchvision, and the detector is trained for 12 epochs with SGD. The learning rate is set to 0.0075 in the three-GPU training setting, with momentum 0.9 and weight decay 10^{-4}. A step learning-rate schedule is used, with decay milestones at epochs 8 and 11. Checkpoints are saved after each epoch. The epoch-6 checkpoint is used in the final system. Its bounding-box detections are converted into temporal events on the validation split, and the post-processing configuration for the retained BAFNet candidates is selected using the corrected one-to-one matching protocol.

For the final BAFNet branch, the original converted training annotations are used without background-only image capping or class-repeat sampling, and random flipping is disabled because the time and frequency axes of spectrograms have fixed physical meanings. No evaluation-set annotations are used for model training, checkpoint selection, or post-processing parameter tuning.

Table 4: Validation results of the selected detection branches.

Model	Recall	Precision	F1
BAFNet	0.657457	0.665728	0.661567
YOLO ensemble	0.684178	0.593095	0.635389
class-wise fusion	0.693084	0.654330	0.673150

4. MAIN RESULTS AND ANALYSIS

Table 4 reports the validation results of the two selected detection branches. The BAFNet branch achieves better validation performance on the selected *bmabz* candidates, with an F1 score of 0.6616. The YOLO ensemble branch achieves a micro F1 score of 0.6339 and is selected for the *d* and *bp* categories according to the class-aware validation strategy. These results support the use of heterogeneous detector branches rather than a single detector for all whale-call categories.

The BAFNet branch provides stronger validation performance for *bmabz*, while the YOLO ensemble branch provides complementary detections for *d* and *bp*. After branch selection, all retained candidates are further refined by the unified event-level post-processing pipeline described above.

5. CONCLUSION

This report describes our BioDCASE 2026 Task 2 whale-call detection system. We formulate the task as spectrogram-based event detection and combine a YOLO ensemble branch with a BAFNet branch. According to validation performance, BAFNet is used for *bmabz*, while the YOLO ensemble branch is used for *d* and *bp*.

A unified event-level post-processing pipeline is applied to refine the selected candidates. It reduces duplicate predictions, compensates for conservative temporal boundaries, recovers weak candidates, and generates a stable official CSV output. Validation results show that the selected BAFNet branch achieves an F1 score of 0.6616, while the YOLO ensemble branch achieves a micro F1 score of 0.6354.

6. REFERENCES

[1] B. S. Miller *et al.*, “An open access dataset for developing automated detectors of Antarctic baleen whale sounds and performance evaluation of two commonly used detectors,” *Scientific Reports*, vol. 11, p. 806, 2021, doi: 10.1038/s41598-020-78995-8.

[2] E. Schall, I. I. Kaya, E. Debusschere, P. Devos, and C. Parcerisas Serrahima, “Deep learning in marine bioacoustics: a benchmark for baleen whale detection,” *Remote Sensing in*

Ecology and Conservation, vol. 10, no. 5, pp. 642–654, 2024, doi: 10.1002/rse2.392.

[3] E. Schall and C. Parcerisas Serrahima, "A robust method to automatically detect fin whale acoustic presence in large and diverse passive acoustic datasets," *Journal of Marine Science and Engineering*, vol. 10, no. 12, p. 1831, 2022, doi: 10.3390/jmse10121831.

[4] D. Stowell, "Computational bioacoustics with deep learning: a review and roadmap," *PeerJ*, vol. 10, p. e13152, 2022, doi: 10.7717/peerj.13152.

[5] J. Song, M. Zhou, J. Luo, H. Pu, Y. Feng, X. Wei, and W. Jia, "Boundary-aware feature fusion with dual-stream attention for remote sensing small object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–13, Art. no. 5600213, 2025, doi: 10.1109/TGRS.2024.3514376.

[6] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10208–10219, doi: 10.1109/CVPR46437.2021.01008.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[8] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019, doi: 10.48550/arXiv.1906.07155.

[9] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. European Conference on Computer Vision*, 2014, pp. 740–755, doi: 10.48550/arXiv.1405.0312.

[10] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th International Conference on Pattern Recognition (ICPR)*, vol. 3, 2006, pp. 850–855, doi: 10.1109/ICPR.2006.479.