WhaleVAD-BPN

Improving Baleen Whale Call Detection with Boundary Proposal Networks and Post-processing Optimisation

Christiaan M. Geldenhuys
cmgeldenhuys@sun.ac.za

Günther Tonitz
gtonitz@sun.ac.za

Thomas R. Niesler
trn@sun.ac.za

Department of Electrical and Electronic Engineering
University of Stellenbosch
Stellenbosch, South Africa

***Abstract*—While recent sound event detection (SED) systems can identify baleen whale calls in marine audio, challenges related to false positive and minority-class detection persist. We propose the boundary proposal network (BPN), which extends an existing lightweight SED system. The BPN is inspired by work in image object detection and aims to reduce the number of false positive detections. It achieves this by using intermediate latent representations computed within the backbone classification model to gate the final output. When added to an existing SED system, the BPN achieves a 16.8 % absolute increase in precision, as well as 21.3 % and 9.4 % improvements in the F1-score for minority-class d-calls and bp-calls, respectively. We further consider two approaches to the selection of post-processing hyperparameters: a forward-search and a backward-search. By separately optimising event-level and frame-level hyperparameters, these two approaches lead to considerable performance improvements over parameters selected using empirical methods. The complete WhaleVAD-BPN system achieves a cross-validated development F1-score of 0.475, which is a 9.8 % absolute improvement over the baseline.**

***Index Terms*—Sound Event Detection, Computational Bioacoustics, Baleen Whale Call Detection, Marine Bioacoustics, Boundary Proposal Network, Post-Processing**

I. Introduction

Passive acoustic monitoring (PAM) has become a cornerstone for assessing marine mammal populations in remote and otherwise inaccessible habitats, owing to its non-invasive nature and relatively modest financial cost. In practice, however, PAM generates substantial volumes of data, often suffering from low signal-to-noise ratios in recordings [1]. Both factors render manual annotation labour-intensive and dependent on specialist knowledge. Consequently, a large body of research has focused on the development of automated detectors for the vocalisations of key species, most notably *Balaenoptera musculus* (blue) and *B. physalus* (fin) whales. These species remain classified as endangered and vulnerable by the IUCN [2], [3], and continue to elude precise abundance estimates because of a scarcity of labelled data [4].

Recent sound event detection (SED) systems have made considerable strides in identifying baleen whale calls within marine recordings, yet these models still exhibit a high false-positive rate and do not allow the reliable detection of infrequent (minority-class) call events [5]. To address these shortcomings, we propose the inclusion of a complementary neural module, the boundary proposal network (BPN). Inspired by the field of object detection in image processing [6], [7], the BPN is intended to supplement an existing lightweight SED architecture by exploiting intermediate latent features already calculated within the backbone classifier. The BPN output is used as a gating mechanism that refines the temporal localisation of detected events, thereby reducing false positives and improving overall precision. It was found that the BPN also leads to an improvement in recall for minority-class call types.

In addition to the described architectural augmentation, we introduce improvements to the post-processing stage of the SED architecture itself. We investigate two search strategies for optimising the hyperparameter selection of the post-processing stage: a forward-search and a backward-search. These procedures systematically explore event- and frame-level hyperparameters, yielding performance gains that surpass those obtained through existing ad hoc or empirical methods.

II. Background

This section provides a summary of the background regarding baleen whale call detection, proposal networks, and the post-processing techniques used.

A. Baleen whale call activity detection

Baleen whales (*mysticetes*) undertake extensive migrations and thus communicate using low-frequency vocalisations that propagate over great distances underwater. Previous work has shown that voice activity detection (VAD) algorithms can be applied to baleen whale call detection, specifically detecting the calls of blue and fin whales [5]. For example, the AVA-VAD [8] system relied on producing latent features from a spectrogram representation of the audio recording, using a convolutional neural network (CNN). These latent features are then processed sequentially by a bidirectional long short-term memory network (BiLSTM) architecture with sigmoid activation to obtain the final posterior classification probabilities for each frame in the input spectrogram.

While phase information has been disregarded in speech processing systems, its inclusion has been reported to afford a 10 % improvement in the F1-score [5]. This work proposed further changes to the AVA-VAD architecture, with the addition of bottleneck and depthwise convolutional layers with recurrent connections. The final proposed system was called WhaleVAD, and it is this system that we will extend.

B. Proposal networks

Proposal networks are a two-stage architecture common in the computer vision field for achieving object detection [6], [9], [10]. These networks have been refined over time to be computationally efficient and have proven to be effective at localising and identifying objects within an image. Early approaches, such as R-CNN (regions with CNN features) [9], relied on external algorithms, such as *selective search* [11], to generate a set of candidate regions of interest (ROIs). These ROIs could then be used by a secondary detection network to determine if a particular object is present in the proposed region. However, these proposal networks relied on handcrafted convolutional features.

Ren, He, Girshick, *et al.* [6] were the first to introduce the region proposal network (RPN), which integrated proposal generation directly into the neural network architecture. The technique relied on applying a small trainable CNN to the feature map of a convolutional backbone network to coordinate and refine a set of predefined anchor boxes. By sharing convolutional features with the downstream detection network, the RPN enabled an efficient and end-to-end trainable system. This architecture established the foundation for most subsequent two-stage object detectors, demonstrating the effectiveness of backbone features.

Despite the success of RPNs, its reliance on a single, deep feature map presents limitations in detecting objects across varying scales and remains computationally less efficient when compared to single-stage detectors. Liu, Anguelov, Erhan, *et al.* [7] address this by attaching multiple detection heads to intermediate layers at varying depths within the backbone network. Other single-stage detectors, such as YOLOv3 [12], adopted a multi-scale feature pyramid network-inspired (FPN) [13] mechanism to aggregate intermediate features.

C. Post-processing

During per-frame SED in a bioacoustic system, the classification model produces a sequence of probabilities over time, each indicating the likelihood of a particular sound class being present in a frame. These *model probabilities* are converted to *binary detections* by applying a threshold to each of the per-frame call probabilities. These binary detections are then aggregated into *discrete events* representing contiguous periods of (call) activity.

The ideal system output is a single event that accurately aligns with the human annotation. In practice, however, these output events are often fragmented, containing intermittent gaps, or spurious or excessively long detections.

To mitigate this, filtering is applied during post-processing. We consider two classes of filtering techniques, namely frame- and event-level.

1) Frame-level techniques: These methods serve to smooth either the model output probabilities, or the detections, or both. For example, a median filter can be applied to the model output, where each probability is replaced by the median value within a neighbourhood. This reduces sporadic peaks or dips in activity, thereby decreasing the number of fragmented events, at the cost of reduced precision. Hysteresis is another common frame-level post-processing approach, where a window of past estimates influences the current model decision. One implementation applies different thresholds for entering and exiting an event state, which we will refer to as *threshold hysteresis* [14]. By setting a lower threshold for termination, the system remains active for longer, which may help reduce event fragmentation. Alternatively, activity at a given time instant can be defined based on the majority vote (statistical mode) within the sliding window, which we will refer to as *hangover*. This can be interpreted as a variant of median filtering applied to model binary detections, as opposed to model class probabilities; given by the following equation:

$$\tilde{y}_t = \begin{cases} 1 & \text{if } \sum_{i=0}^{k} \hat{y}_{t-i} > \dfrac{k+1}{2}, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $\tilde{y}_t$ is the model detection at time instant t, and k is the number of past samples in the sliding window.

2) Event-level techniques: These methods act at an event-level on the aggregated detections and commonly impose constraints on duration. Minimum and maximum event durations may be defined to remove events of implausible duration, and a minimum inter-event duration may be defined to merge events occurring close in time.

The effectiveness of these post-processing methods depends on a set of hyperparameters, such as the size of the neighbourhood used to implement median filtering or hangover. The selection of these hyperparameters is highly task and data-specific, and thus requires optimisation.

III. Literature Review

This section starts with an overview of previous research in the detection and classification of baleen whales, followed by a focus on post-processing strategies that have been implemented and evaluated in bioacoustic SED.

A. Detection and classification of baleen whale calls

Automated methods for the detection and classification of baleen whales continue to develop. Due to the great distances over which their vocalisations propagate underwater, these calls are an ideal candidate for PAM [15].

Before the advent of machine learning, automated detection relied on classical signal-processing techniques designed to identify signals with known, empirical characteristics. Two common approaches were matched filtering, in which a synthetic kernel derived from a known call is correlated with a recording to locate vocalisations in background noise [15], and spectrogram correlation, which cross-correlates a template spectrogram with successive recording segments to identify vocal occurrences [16].

Machine learning approaches were subsequently adopted to address the non-stationary nature of ocean soundscapes and reduce the reliance on fixed filters or templates. Examples include support vector machines (SVMs) to identify individual humpback whale vocalisations [17] and North Atlantic right whale upcalls [18], as well as probabilistic models such as Gaussian mixture models (GMMs) for blue whale calls [19] and hidden Markov models (HMMs) for humpback whale calls [20].

More recently, deep neural networks (DNNs), particularly CNNs, have become a common area of research for computational bioacoustics. Large CNN architectures, such as DenseNet [21], have been applied to baleen whale classification; for example, Miller, Madhusudhana, Aulich, *et al.* [22] reported that a DenseNet-based system surpassed human observers in blue whale D-call detections. Convolutional recurrent neural networks (CRNNs), which integrate recurrent layers to capture temporal dependencies, have also shown strong performance in fin- and blue whale classification [23]. Despite continued improvements in frame-level detector performance [24], lower false-positive rates are desirable, as frame-level errors can cascade into greater event-level errors.

B. Post-processing in bioacoustic SED

Frame-level post-processing commonly involves class-dependent thresholds [4], [25], [26] and median filters with a fixed kernel size [26]. Hoffman and Robinson [25] report that class-specific thresholds, applied to logits, can outperform a fixed threshold on bird-call detection. Hysteresis thresholds have not been reported in the bioacoustic literature. However, this technique has been applied in other SED domains. For example, Cances, Pellegrini, and Guyot [27] found hysteresis thresholds to outperform absolute thresholds when classifying sound classes that occur in a domestic environment. Hangover has also not been reported on in SED studies, although it has been applied successfully in other detection problems, such as sonar [28] and radar [29], where an *M-out-of-N* criterion is used.

Event-level post-processing techniques are common in few-shot bioacoustic learning. For example, several studies impose a minimum event duration and a minimum inter-event interval. These hyperparameters are commonly derived from the support set [25], [30]. Miller *et al.* [4] apply a minimum and maximum event duration of 0.5 s and 2.5 s respectively, and require at least 0.5 s between events when identifying fin whale 20 Hz pulses.

For domestic environmental SED, Cances, Guyot, and Pellegrini [14] have provided a comprehensive comparison of frame-level post-processing techniques, including fixed threshold, hysteresis threshold, and a slope-based threshold technique which detects fast changes in model probabilities. Each of the threshold techniques were evaluated using class-dependent and class-independent hyperparameters. The authors report an 28.6 % absolute improvement in the F1-score over the baseline, when selecting class-dependent fixed thresholds. This suggests that the selection of class-specific hyperparameters and the use of multiple post-processing strategies in bioacoustic SED may yield substantial gains in detection performance. However, to our knowledge, no bioacoustic SED study has reported a systematic treatment of different post-processing strategies.

IV. Data

As part of the 2025 IEEE BioDCASE (Task 2) challenge, a dataset consisting of strongly labelled blue- and fin-whale calls in the Antarctic region of the Southern Ocean was released [31]. The data was originally obtained by the Antarctic Blue and Fin Whale Acoustic Trends Project (ATP) as part of the International Whaling Commission's Southern Ocean Research Partnership (IWC-SORP) [32].

The Acoustic Trends Blue Fin Library (ATBFL) consists of 11 site-year datasets recorded around the Antarctic, in the period 2005 to 2017. Sites were selected based on the availability of a full year of data, and utilised different recording instrumentation. All audio data was resampled to a rate of 250 Hz. Each dataset was manually annotated, in both the time and frequency domains, by domain experts using the data collection and annotation procedures described in Miller *et al.* [4].

The challenge identifies three site-year datasets as a development set, while the remaining eight site-year sets form the training set. The entire dataset contains a total of 6594 recordings with a total duration of 76 178 hours, as set out in Table I. Only 5.2 % of the data (in duration) contains blue or fin whale vocalisations, however.

The ATBFL includes annotations for seven different call types, of which four are produced by blue whales (*BmA*, *BmB*, *BmZ* and *BmD*) and three by fin whales (*BpD*, *Bp20* and *Bp20plus*). A blue whale Z-call (*BmZ*) is a low-frequency compounded call consisting of both the A-call (*BmA*) and the B-call (*BmB*). The blue whale D-call (*BmD*) is a downsweeping call, ranging between 20 Hz to 120 Hz, which is similar to a fin whale downsweep (*BpD*) which ranges from 30 Hz to 90 Hz. Finally, fin whales also create a downsweeping pulse between 15 Hz to 30 Hz which can either appear with (*Bp20plus*) or without (*Bp20*) an overtone varying between 80 Hz to 120 Hz. Figure 1 shows examples of each call type.

Following Schall, Kaya, Debusschere, *et al.* [24], call types are grouped based on their acoustic similarity and interrelated usage. The A-, B-, and Z-calls, which co-occur, are combined into a single ABZ-call category (`bmabz`). Similarly, the blue whale D-call and the fin whale BpD-call are grouped into a unified D-call category (`d`), while the Bp20 and Bp20Plus calls are merged into the Bp-call category (`bp`). These call grouped categories serve as the final call labels used.

Table II summarises the duration, frequency characteristics, and annotation counts for all call types found in the training and the development sets. Blue whale vocalisations are more common in both datasets, comprising 74.8 % of the training set and 65.9 % of the development set annotations. Notably,

TABLE I: Summary of the ATBFL site-year training and development sets. The table shows the average recording duration (hours), number of recordings, total recording duration (hours), number of annotated events, and total duration of whale calls (hours) for each set.

Dataset	Avg. Duration (h)	Recordings	Total duration (h)	Total events	Total event duration (h)
ballenyisland2015	1.0	205	204	2222	2.8
casey2014	1.0	194	194	6866	14.2
elephantislands2013	0.08	2247	187	21949	16.1
elephantislands2014	0.08	2595	216	20962	28.1
greenwich2015	0.17	190	32	1128	2.1
kerguelen2005	1.0	200	200	2960	3.5
maudrise2014	0.42	200	83	2360	5.7
rosssea2014	1.0	176	176	104	0.1
Total training set	–	**6007**	**1292**	**58551**	**72.6**
casey2017	1.0	187	185	3263	6.1
kerguelen2014	1.0	200	200	8822	11.4
kerguelen2015	1.0	200	200	5542	7.4
Total development set	–	**587**	**585**	**17627**	**24.9**

TABLE II: Different whale call frequency (hertz) and duration (seconds) information computed from data in [31].

(a) Training Set

Type	Frequency (Hz)			Duration (s)			Count
	Min	Max	Avg.	Min	Max	Avg.	
BmA	11.4	110.6	25.9	2.12	27.11	7.19	13785
BmB	10.0	31.3	22.2	3.14	19.51	7.83	5433
BmZ	11.5	34.6	22.0	3.87	28.07	12.76	1646
BmD	11.5	110.7	69.9	0.29	6.78	1.42	22977
BpD	16.7	134.1	75.4	0.29	2.70	1.12	2658
Bp20	8.5	45.1	22.2	0.48	3.08	1.52	9104
Bp20plus	9.2	112.7	52.5	0.76	2.91	1.50	3950

(b) Development Set

Type	Frequency (Hz)			Duration (s)			Count
	Min	Max	Avg.	Min	Max	Avg.	
BmA	15.7	30.1	25.6	2.12	36.62	7.12	6268
BmB	10.7	99.0	22.6	1.29	18.1	8.35	2277
BmZ	12.1	30.3	21.9	5.15	29.45	12.64	918
BmD	15.9	122.9	57.4	0.74	7.36	2.87	2168
BpD	26.5	137.5	61.7	0.37	2.58	1.08	688
Bp20	10.3	47.9	22.7	0.46	2.83	1.35	2550
Bp20plus	11.1	106.6	57.1	0.64	2.58	1.43	2758

blue whale A-calls, B-calls, and Z-calls exhibit longer call durations, accounting for $45.18\,\mathrm{hours}$ in the training set and $20.9\,\mathrm{hours}$ in the development set. This distribution reveals an imbalance, both in species representation (blue whales are overrepresented) and in the temporal occurrence of call types.

V. Experimental Structure

In the following, we first describe the architectural changes we have made to the WhaleVAD system. Then, we present our boundary proposal network, which used intermediate features from within the WhaleVAD system to improve overall model performance. Finally, we present two post-processing optimisation search strategies that separately optimise frame-level and event-level hyperparameters.

A. Architecture modifications

In the original WhaleVAD model [5], the *depthwise convolution* block consisted of three depthwise convolution layers placed in series. We adapt the depthwise convolution block by adding residual connections between each of the layers with increasing dilation of 2, 4 and 8. The increase in dilation factor provides a wider receptive field, thus allowing the network to utilise features that are further away in time [33]. Each depthwise convolution block retains the GELU activation and batch normalisation used by the original model. In addition, the conventional dropout has been replaced with spatial dropout [34], which has been shown to improve the effective regularisation for convolutional layers.

B. Boundary proposal network

While the original WhaleVAD model architecture achieved high recall, the model also exhibited a high false positive rate (FPR), which in turn resulted in a low precision [5]. We propose a new component, the boundary proposal network (BPN), whose purpose is to compute a gating score that is combined with the WhaleVAD classifier output in an effort to reduce false positives. The boundary proposal network (BPN) uses convolutional feature representations obtained from intermediate layers, referred to as *intermediate feature maps*, from the backbone WhaleVAD classifier, to compute this gating score.

The following subsections describe the components that comprise the BPN, as well as the training regime used.

1) Intermediate projection head: Each set of intermediate feature maps is processed by a separate CNN, called an intermediate projection head. This consists of a convolutional layer with batch normalisation, GELU activation and a final maximum pooling layer. Each projection head shares the same architecture, but has its own set of weights associated with a particular point in the backbone network, from which the feature map was drawn.

2) Proposal network: The outputs of all heads are concatenated along a new dimension H, which is processed by the

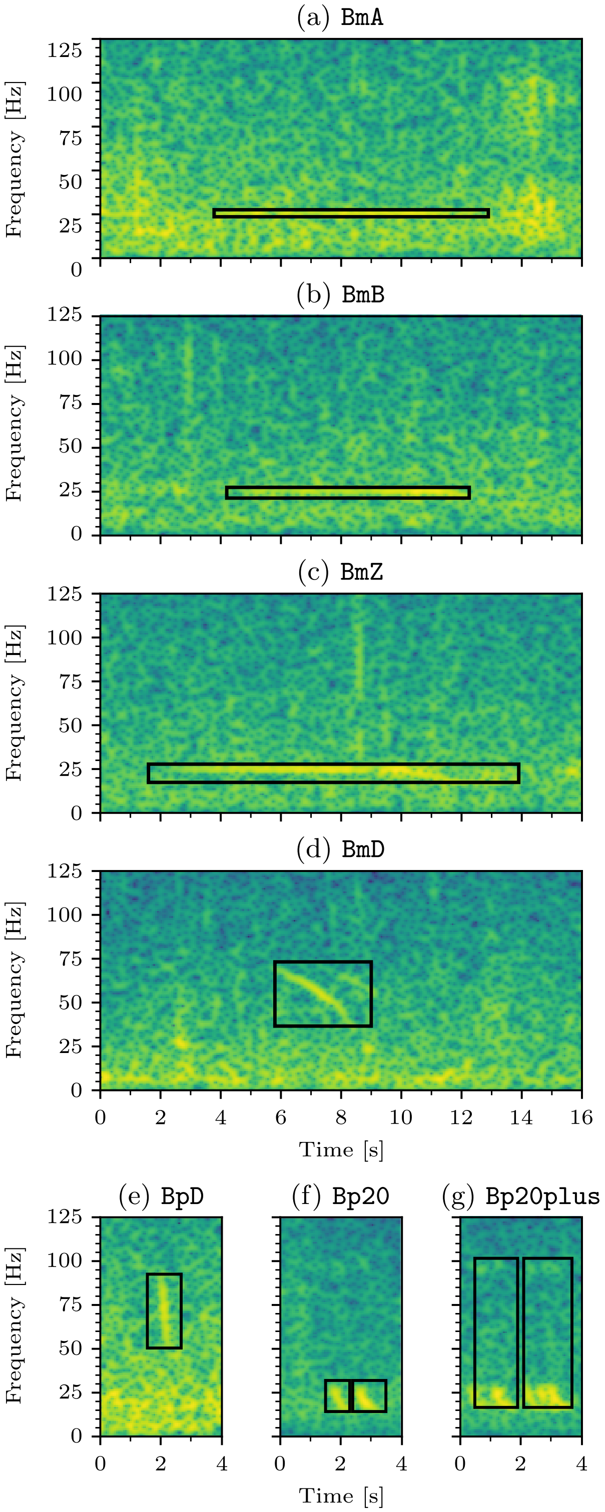


Fig. 1: Spectrogram representations of exemplar baleen whale call types [31]. Bounding boxes indicate presence of particular call type, provided by domain expert annotations in dataset. Figures (a) – (d) correspond to blue whale vocalisations, and (e) – (g) correspond to fin whale vocalisations.

TABLE III: The boundary proposal network (BPN) layerwise configuration in terms of kernel size (K), stride (S), number of input channels (C_{in}) and output channels (C_{out}).

Layer	K	S	C_{in}	C_{out}
Intermediate projection head				
└ Conv2D	(1, 1)	(1, 1)	128	128
└ Max pool	(3, 1)	(1, 1)	–	–
Proposal network				
└ Conv2D transpose	(4, 1)	(1, 1)	128	128
└ Conv2D transpose	(5, 1)	(1, 1)	128	64

proposal network to produce R distinct ROI vectors per head. Each ROI consists of a latent feature vector with dimensionality C_{bpn}, associated with a projection head ($h_i \in H$) for each time instant T. The proposal network consists of two convolutional layers with GELU activation, batch normalisation and spatial dropout [34]. We evaluate two variants of the network. *BPN-multi* produces multiple ROIs per projection head using a convolutional transpose ($R > 1$), whilst *BPN-single* produces a single ROI per projection head ($R = 1$). Early experimentation showed that BPN-multi outperformed BPN-single and thus only BPN-multi was considered in the final results, and will be referred to as simply BPN throughout. Table III shows a summary of the BPN layer configuration.

3) BiLSTM: Each latent ROI vector is processed sequentially in time by a BiLSTM or independently by a logistic regression (LR) module; with a sigmoid activation on the output. During hyperparameter optimisation, it was found that the BiLSTM network outperforms LR. The resulting output is averaged over each ROI using a learned weighted mean to produce a *mask*. The weighted average is jointly trained with the model, resulting in some heads being weighted more heavily than others. This weighting remains fixed during inference.

4) Masking: The final mask is applied to the posterior call probabilities produced by the backbone classifier, thus acting as a soft gating mechanism whose purpose is to suppress spurious detections (false positives). As a result, the final posterior call probabilities are dependent on both the classification network, which is responsible for localising and identifying a particular call type in time, and the gating mechanism of the BPN which aligns with this postulated call. The addition of the BPN gating mechanism should therefore allow the number of false positive classifications made by the original WhaleVAD architecture to be reduced through training. We will refer to our model as WhaleVAD-BPN. Figure 2 shows an illustration of the complete WhaleVAD-BPN system.

5) Training regime: All models were trained using the AdamW [35] optimiser with Focal loss [36]. The training set was divided into mini-batches of 48 segments per batch, each consisting of approximately 30 seconds long. The learning rate is kept fixed at 0.001 with momentum terms of 0.9 and 0.999 and a weight decay factor of 0.01. Training is halted once the training loss has converged or after 32 epochs over the entire training set.

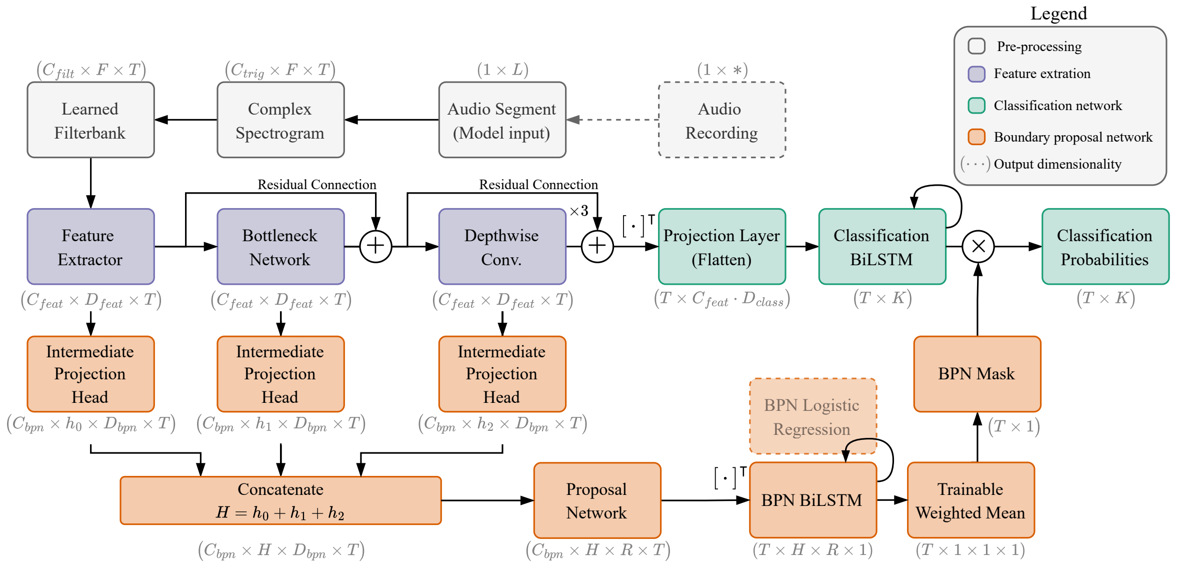

Legend
Pre-processing
Feature extration
Classification network
Boundary proposal network
$(\cdots)$ Output dimensionality
$(C_{filt} \times F \times T)$
Learned Filterbank
$(C_{trig} \times F \times T)$
Complex Spectrogram
$(1 \times L)$
Audio Segment (Model input)
$(1 \times *)$
Audio Recording
Feature Extractor
Residual Connection
Bottleneck Network
Residual Connection
Depthwise Conv.
×3
$[\cdot]^{\intercal}$
Projection Layer (Flatten)
Classification BiLSTM
Classification Probabilities
$(C_{feat} \times D_{feat} \times T)$
$(C_{feat} \times D_{feat} \times T)$
$(C_{feat} \times D_{feat} \times T)$
$(T \times C_{feat} \cdot D_{class})$
$(T \times K)$
$(T \times K)$
Intermediate Projection Head
Intermediate Projection Head
Intermediate Projection Head
BPN Mask
BPN Logistic Regression
$(C_{bpn} \times h_0 \times D_{bpn} \times T)$
$(C_{bpn} \times h_1 \times D_{bpn} \times T)$
$(C_{bpn} \times h_2 \times D_{bpn} \times T)$
$(T \times 1)$
Concatenate $H = h_0 + h_1 + h_2$
Proposal Network
$[\cdot]^{\intercal}$
BPN BiLSTM
Trainable Weighted Mean
$(C_{bpn} \times H \times D_{bpn} \times T)$
$(C_{bpn} \times H \times R \times T)$
$(T \times H \times R \times 1)$
$(T \times 1 \times 1 \times 1)$

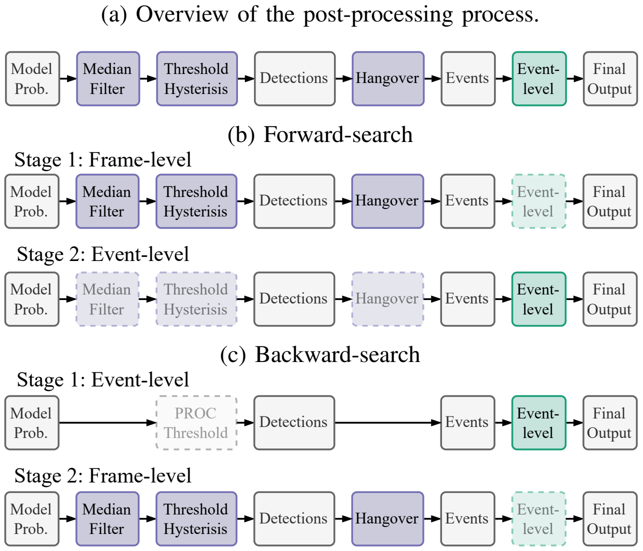
Fig. 2: Illustration of the WhaleVAD-BPN system we propose for improved whale call boundary detection. The system is divided into four sections: pre-processing, feature extraction, classification, and boundary proposal. The output tensor dimensionality is shown below each module in the system.

C. Post-processing hyperparameter selection

During the evaluation of the different post-processing techniques, we employ a three-fold cross-validation scheme using the development sets [37]. For cross-validation, the data is partitioned into disjoint sets, referred to as *folds*. The development set consists of three site-year subsets, each of which is assigned to a different fold. The best post-processing hyperparameters are chosen based on the highest F1-score over two of the folds (development folds), while the third fold is held out for testing (test fold). After the parameters are chosen based on the development folds, the system is evaluated on the test fold. The chosen development and test folds are permuted, referred to as a *turn*, and the process is repeated. The final system evaluation is computed by averaging the score over all three turns of the test folds.

Each post-processing hyperparameter is either applied at a frame-level or at an event-level, as discussed in Section II-C. We propose two selection procedures, namely *forward-search* and *backward-search*. During either search method, both frame-level and event-level hyperparameters are searched separately using a two-stage process. An exhaustive optimisation across both hyperparameter types was practically infeasible, as it would require either substantial computational resources or a large reduction in the search space.

1) Forward-search: During the forward-search (refer to Fig. 3b), event-level hyperparameters are initially fixed (Stage 1) based on the statistical properties derived from the dataset. The minimum inter-event duration is fixed at 500 ms. For each of the three target classes, the minimum and maximum event durations are drawn from the ranges in Table II by taking, within each group, the overall minimum and maximum of the constituent classes. The frame-level hyperparameters are searched based on Table IVa: median filter kernel size,


(a) Overview of the post-processing process.

(b) Forward-search

(c) Backward-search

Fig. 3: Illustration of the post-processing applied to the call probabilities produced by a per-frame SED model. (a) An overview of the steps employed during both frame-level (*purple*) and event-level (*green*) post-processing. (b and c) The forward- and backward-search approaches to the selection of post-processing hyperparameters, respectively. Both optimisation methods consist of two stages, where some parts of the network remain fixed (opaque dashes) while hyperparameters for the remainder are searched to maximise the F1-score over all development folds.

TABLE IV: Ranges of the hyperparameter values considered for (a) frame-level, and (b) event-level post-processing.

(a) Frame-level hyperparameters.

Parameter	Search space
Filter kernel	[None, 11, 33, 55]
On threshold	0.1 → 0.9 (inc. 0.1)
Off threshold	0.1 → 0.9 (inc. 0.1)
Hangover kernel	[None, 11, 33, 55]

(b) Event-level hyperparameters.

Parameter	bmabz	d	bp
Min. time between events (s)	0.1 → 0.9 (inc. 0.1)	0.1 → 0.9 (inc. 0.1)	0.1 → 0.9 (inc. 0.1)
Min. event duration (s)	2.0 → 5.0 (inc. 0.5)	0.6 → 3.0 (inc. 0.4)	0.3 → 1.5 (inc. 0.2)
Max. event duration (s)	25.0 → 40.0 (inc. 2.5)	5.0 → 11.0 (inc. 1.0)	2.0 → 5.0 (inc. 0.5)

threshold, hysteresis (off) threshold, and the hangover kernel size. Note that each of these hyperparameters is class-dependent. Next, in Stage 2, these candidate frame-level hyperparameters are fixed, while the event-level hyperparameters are searched based on Table IVb.

2) *Backward-search:* During the backward-search (refer to Fig. 3c), frame-level hyperparameters (thresholds) are initially fixed (Stage 1) based on equal precision-recall, obtained from the average precision-recall curve computed over the development folds. Event-level hyperparameters are searched based on Table IVb. Next, in Stage 2, these candidate event-level hyperparameters are fixed, while the frame-level hyperparameters are considered based on Table IVa.

3) *Final model evaluation:* After the hyperparameters have been fixed using either search method, precision and recall metrics are computed for each test fold. The final F1-score, per call type, is then recomputed from the averaged precision/recall scores. The final system evaluation is based on the macro F1-score of each class, again using the averaged precision/recall scores [31]. The hyperparameter selection process is repeated for both the WhaleVAD and the WhaleVAD-BPN models.

VI. Results

Table V shows the cross-validated performance of the baseline model when applying hyperparameter selection, using backward-search, on a per-class basis. Applying event-level optimisation alone led to consistent performance gains for all classes, when compared to the previously-used empirical method of selecting the post-processing parameters (Table VI). In particular, for D-calls (d) we see a 9.3 % absolute improvement in the F1-score, which corresponds to a 77.5 % relative improvement over the empirically derived parameters. Additional frame-level optimisation provides more modest gains, leading to a final macro F1-score of 0.422 across all classes. Overall, this represents a 4.5 % absolute improvement in the macro F1-score, compared to the baseline WhaleVAD model, achieved solely through the optimisation of post-processing and no architectural modifications (see Table VII). Forward-search achieved similar performance to backward-search. However, since backward-search performed slightly better, results for only this method will be shown.

Figure 4 shows that the WhaleVAD-BPN architecture yields superior precision-recall curves across all three call types relative to the original WhaleVAD system, indicating better performance even without optimisation of the post-processing. The model successfully leads to improved precision (fewer false positives) across a broad range of classification thresholds and particularly for the minority classes d and bp.

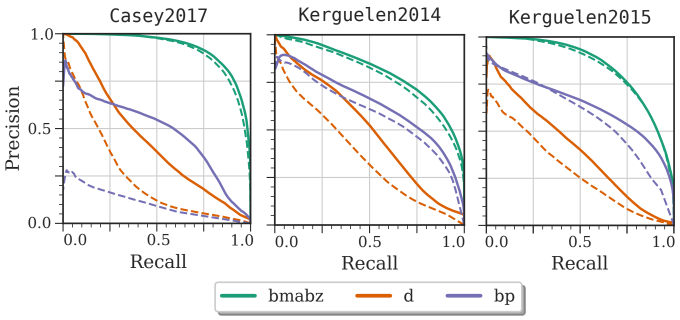


Fig. 4: Precision-recall curves for each call type and for each of the three site-year development sets. Both the baseline WhaleVAD (dashed) and the proposed WhaleVAD-BPN (solid) models are shown.

The application of hyperparameter optimisation using backward-search for the proposed WhaleVAD-BPN model is shown in Table V. Frame-level optimisation achieves substantial improvements and is more successful than event-level optimisation. Specifically, bmabz-, d- and bp-calls improve by 6.9 %, 9.8 %, and 4.8 %, respectively. The final macro F1-score reaches 0.475, representing a 9.8 % absolute improvement over the original WhaleVAD model and a 5.3 % improvement over WhaleVAD with optimised post-processing (see Table VII).

To investigate the individual contribution of each frame-level post-processing technique, we also evaluated the absolute improvement when each technique is applied in addition to selecting class-dependent thresholds. Specifically, we consider class-dependent hysteresis thresholds, hangover, and median filtering. Although a few cases yielded noticeable gains, most contributed relatively little compared to class-dependent threshold optimisation alone. As shown in Table V, event-level hyperparameter optimisation produces large performance gains, challenging the prevalent current practice of simply deriving these values from dataset statistics.

Finally, Table VII highlights the trade-offs underlying these gains. The optimised WhaleVAD model improves the F1-score primarily by sacrificing recall in exchange for higher precision. By contrast, WhaleVAD-BPN manages to achieve a marginal improvement in recall compared to the baseline WhaleVAD model whilst also achieving a substantial increase in precision. Therefore, the boundary proposal network has succeeded in reducing false positives while preserving recall.

TABLE V: Cross-validation F1-score for WhaleVAD (baseline) and WhaleVAD-BPN models using a backward-search for selecting the post-processing hyperparameters. (a, Stage 1) Event-level: classification thresholds are selected from the average precision-recall curve over the development folds, after which event-level hyperparameters are selected. (b, Stage 2) Frame-level: hyperparameters are selected, while event-level hyperparameters remain fixed at the previously selected values. The final F1-score is recalculated based on the average of the precision and recall for each test fold.

(a) Stage 1: Event-level selection

Call type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
				WhaleVAD			
bmabz	0.634	0.686	0.665	0.620	0.670	0.591	**0.663**
d	0.212	0.162	0.226	0.186	0.199	0.220	0.219
bp	0.529	0.029	0.291	0.446	0.237	0.560	0.348
				WhaleVAD-BPN			
bmabz	0.546	0.510	0.536	0.644	0.526	0.410	0.546
d	0.238	0.233	0.204	0.152	0.235	0.307	**0.242**
bp	0.512	0.288	0.376	0.451	0.339	0.489	**0.412**

(b) Stage 2: Frame-level selection

Call type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
				WhaleVAD			
bmabz	0.626	0.698	0.659	0.632	0.665	0.621	**0.669**
d	0.209	0.223	0.233	0.175	0.199	0.242	0.222
bp	0.530	0.029	0.302	0.485	0.257	0.575	0.363
				WhaleVAD-BPN			
bmabz	0.590	0.603	0.585	0.614	0.608	0.567	0.615
d	0.311	0.388	0.377	0.256	0.322	0.366	**0.340**
bp	0.532	0.311	0.442	0.492	0.401	0.573	**0.460**

TABLE VI: Cross-validation F1-score for WhaleVAD when no frame-level post-processing is applied and event-level hyperparameters are selected from empirical call statistics.

Type	Fold 1		Fold 2		Fold 3		Final F1
	Dev	Test	Dev	Test	Dev	Test	
bmabz	0.615	0.620	0.620	0.657	0.631	0.536	0.628
d	0.136	0.049	0.106	0.128	0.099	0.168	0.126
bp	0.494	0.038	0.286	0.386	0.217	0.521	0.315

TABLE VII: Final results comparing the original WhaleVAD architecture, a variant with post-processing optimised via cross-validation, and the best reported model, WhaleVAD-BPN, also with optimisation. Scores are averaged across call types and test folds.

Model	IoU	Recall	Precision	F1
WhaleVAD [5]	0.588	0.458	0.320	0.377
WhaleVAD + optimisation	0.601	0.391	0.458	0.422
WhaleVAD-BPN + optimisation	**0.625**	**0.463**	**0.488**	**0.475**

VII. Conclusion

In this work, we present a novel and computationally lightweight network augmentation (BPN) for an existing whale call detection system, as well as a computationally tractable approach to hyperparameter selection for the system post-processing. Both innovations are shown to lead to substantial performance improvements over the baseline system using the 2025 DCASE (Task 2) challenge data, for automated baleen whale call detection.

By exploiting intermediate latent features already computed within the main classifier to act as a gating mechanism for the output, the BPN consistently reduces false positive rates across all classes. The augmented architectures have also shown to improve minority-class call detection, which is generally more difficult than the detection of abundant classes. This is important because annotated data is difficult to obtain, and therefore better performance for a small pool of training examples is highly desirable.

We further demonstrate that the principled selection of post-processing hyperparameters has a marked impact on final system performance. We compare two hyperparameter selection strategies, namely a forward- and backward-search, which both achieve comparable gains. When comparing hyperparameters selected in this way to conventional empirical or ad-hoc choices, a $4.5\,\%$ absolute improvement is seen.

Our final system, which includes the proposed BPN and optimised post-processing hyperparameters, achieves a $9.8\,\%$ absolute improvement in overall F1-score. The system succeeds in markedly reducing false positives, while improving the detection of minority-class calls. In addition to maintaining the already strong recall performance of the baseline baleen whale call detection system, these improvements narrow the performance gap between minority-class calls and calls for which there is abundant data. The proposed system should therefore be a useful tool for the discovery and monitoring of new call types, for which data will initially always be limited.

References

[1] K. A. Kowarski and H. Moors-Murphy, “A review of big data analysis methods for baleen whale passive acoustic monitoring,” *Marine Mammal Science*, vol. 37, no. 2, pp. 652–673, 2021, ISSN: 0824-0469, 1748-7692. DOI: 10.1111/mms.12758.

[2] J. G. Cooke, “Balaenoptera physalus,” *The IUCN Red List of Threatened Species*, 2018.

[3] J. G. Cooke, “Balaenoptera musculus,” *The IUCN Red List of Threatened Species*, 2018, Erratum published in 2019.

[4] B. S. Miller, The IWC-SORP/SOOS Acoustic Trends Working Group, K. M. Stafford, *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, no. 1, p. 806, 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-020-78995-8.

[5] C. M. Geldenhuys, G. Tonitz, and T. R. Niesler, "Whale-VAD: Whale vocalisation activity detection," DCASE2025 Challenge, Technical Report, 2025.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS 2015)*, vol. 28, Montreal, Canada, 2015.

[7] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands: Springer, 2016, pp. 21–37.

[8] N. Wilkinson and T. Niesler, "A hybrid CNN-BiLSTM voice activity detector," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada: IEEE, 2021, pp. 6803–6807, ISBN: 978-1-72817-605-5. DOI: 10.1109/ICASSP39728.2021.9415081.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[10] R. Girshick, "Fast r-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[11] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013, ISSN: 1573-1405. DOI: 10.1007/s11263-013-0620-5.

[12] J. Redmon and A. Farhadi, *YOLOv3: An incremental improvement*, 2018. DOI: 10.48550/arXiv.1804.02767. arXiv: 1804.02767[cs].

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2017, pp. 2117–2125.

[14] L. Cances, P. Guyot, and T. Pellegrini, *Evaluation of post-processing algorithms for polyphonic sound event detection*, 2019. DOI: 10.48550/arXiv.1906.06909.

[15] D. K. Mellinger and C. W. Clark, "Methods for automatic detection of mysticete sounds," *Marine and Freshwater Behaviour and Physiology*, vol. 29, pp. 163–181, 1997.

[16] D. K. Mellinger and C. Clark, "Recognizing transient low-frequency whale sounds by spectrogram correlation.," *The Journal of the Acoustical Society of America*, vol. 107, no. 6, pp. 3518–29, 2000. DOI: 10.1121/1.429434.

[17] S. Mazhar, T. Ura, and R. Bahl, "Vocalization based individual classification of humpback whales using support vector machine," in *OCEANS 2007*, 2007, pp. 1–9. DOI: 10.1109/OCEANS.2007.4449356.

[18] A. K. Ibrahim, H. Zhuang, N. Erdol, and A. M. Ali, "A new approach for north atlantic right whale upcall detection," in *2016 International Symposium on Computer, Consumer and Control (IS3C)*, 2016, pp. 260–263. DOI: 10.1109/IS3C.2016.76.

[19] A. Cuevas, A. Veragua, S. Español-Jiménez, G. Chiang, and F. A. Tobar, "Unsupervised blue whale call detection using multiple time-frequency features," in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2017, pp. 1–6. DOI: 10.1109/CHILECON.2017.8229663.

[20] F. Pace, P. R. White, and O. Adam, "Hidden markov modeling for humpback whale (megaptera novaeanglie) call classification," in *Meetings on Acoustics*, vol. 17, 2012. DOI: 10.1121/1.4772751.

[21] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, 2018. DOI: 10.48550/arXiv.1608.06993.

[22] B. S. Miller, S. Madhusudhana, M. G. Aulich, and N. Kelly, "Deep learning algorithm outperforms experienced human observer at detection of blue whale d-calls: A double-observer analysis," *Remote Sensing in Ecology and Conservation*, vol. 9, no. 1, V. Lecours and D. Risch, Eds., pp. 104–116, 2023. DOI: 10.1002/rse2.297.

[23] J. H. Rasmussen and A. Sirovic, "Automatic detection and classification of baleen whale social calls using convolutional neural networks.," *The Journal of the Acoustical Society of America*, vol. 149, no. 5, p. 3635, 2021.

[24] E. Schall, I. I. Kaya, E. Debusschere, P. Devos, and C. Parcerisas, "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, ISSN: 2056-3485, 2056-3485. DOI: 10.1002/rse2.392.

[25] B. Hoffman and D. Robinson, "Toward in-context bioacoustic sound event detection," DCASE2024 CHallenge, Technical Report, 2024.

[26] P. Zhao, C. Lu, and L.-W. Zou, *Few-shot bioacoustic event detection with frame-level embedding learning system*, 2024. DOI: 10.48550/arXiv.2407.10182.

[27] L. Cances, T. Pellegrini, and P. Guyot, "Multi-task learning and post processing optimization for sound event detection," DCASE2019 Challenge, Technical Report, 2019.

[28] B. Barshan and B. Ayrulu, "Performance comparison of four time-of-flight estimation methods for sonar signals," *Electronics Letters*, vol. 34, no. 16, pp. 1616–1617, 1998. DOI: 10.1049/el:19981127.

[29] J. DiFranco and W. Rubin, *Radar detection* (The Artech radar library). Artech House, 1980, ISBN: 978-0-89006-092-6.

[30] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, *Surrey system for DCASE 2022 task 5: Few-shot bioacoustic event detection with segment-level metric learning*, 2022. DOI: 10.48550/arXiv.2207.10547.

[31] L. Jean-Labadye, C. Parcerisas, B. Miller, *et al.*, *BioDCASE 2025 task 2: Development set*, version 1 (Erratum), Zenodo, 2025. DOI: 10.5281/zenodo.15092732.

[32] B. S. Miller, K. M. Stafford, I. Van Opzeeland, *et al.*, *An annotated library of underwater acoustic recordings for testing and training automated algorithms for detecting antarctic blue and fin whale sounds*, version 1, Australian Antarctic Data Centre, 2020. DOI: 10.26179/5e6056035c01b.

[33] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS 2016)*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Barcelona, Spain: Curran Associates, Inc., 2016.

[34] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, 2018. DOI: 10.48550/arXiv.1708.02002. arXiv: 1708.02002.

[37] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, B, vol. 36, no. 2, pp. 111–133, 1974.