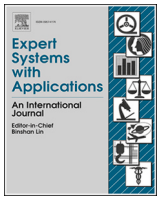
Expert Systems With Applications 331 (2026) 133203



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa





Two-stage acoustic event detection and classification for horn signals in urban scenarios: Comparing VGGish, YAMNet and Mel spectrogram approaches with mRMR feature selection

Vahid Hajihashemi [a], Abdorreza Alavi Gharahbagh [a], José J.M. Machado [b], João Manuel R.S. Tavares [b,*]

[a] Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, Porto, 4200-465, Portugal
[b] Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, Porto, 4200-465, Portugal

ARTICLE INFO

Keywords:
Acoustic event detection
Acoustic embeddings
Feature selection
Hybrid machine learning systems
Temporal smoothing

ABSTRACT

Urban vehicle accidents highlight the need for intelligent systems that can process environmental acoustic signals and improve driver awareness. Traditional acoustic event detection and classification (AEDC) methods often struggle to perform effectively when multiple acoustic events overlap in frequency. This study presents a two-stage framework for detecting and classifying horn signals in urban driving scenarios. The first stage separates horn signals from background noise using Log-Mel spectrogram features and a bagging ensemble classifier. The second stage classifies detected horns into four categories (boat, car, train, truck) using VGGish features. Feature dimensionality is reduced by approximately 40% through minimum Redundancy Maximum Relevance (mRMR) selection without losing accuracy. A post-processing step based on the binomial probability distribution corrects errors across consecutive frames in both stages. The proposed system achieves a minimum F1 score of 0.82 before post-processing and 0.96 after post-processing across all horn classes. This represents more than a 15% improvement over recent deep learning approaches, while maintaining low computational demands through feature selection. The hybrid architecture makes AEDC practical for driver assistance systems in automotive environments with limited processing resources.

1. Introduction

Traffic accidents, particularly vehicle collisions, and their associated economic and human costs (Alemany et al., 2013; Gharahbagh et al., 2021; Moradi, 2025), currently represent one of the most significant challenges in the global transportation industry. The accident rate is particularly significant in developing countries due to inadequate infrastructure and substandard vehicle quality. Despite owning only about 60% of the world's vehicles, these countries account for more than 92% of all road traffic accidents (Organization, 2023). Car accidents occur due to three primary factors: environmental, vehicle-related, and human error. Research across different countries shows that human errors contribute the largest share (Bucsuházy et al., 2020). Hazard detection, the ability to identify hazardous traffic situations, represents a critical driving skill (Crundall et al., 2021; Habibzadeh et al., 2023). Despite advances in related industries, such as Advanced Driver Assistance Systems (ADAS), the percentage of road accidents worldwide remains high. One solution is to enhance artificial intelligence (AI) in existing vehicles to more effectively alert drivers to environmental hazards. The most readily available signals for driver hazard detection are visual and auditory, which AI-based systems can utilise to monitor hazardous situations and provide early warnings to drivers. Current ADAS architectures predominantly employ radar and camera sensor fusion with deep neural network (DNN) processing for lane departure warnings (Mandal et al., 2021; Wang & Lu, 2024), drowsiness monitoring (Albadawi et al., 2022), and driver stress assessment (Wan et al., 2025; Zhao et al., 2024). However, these vision-based systems are unreliable in adverse conditions, such as fog, nighttime operation, and occlusions in the field of view.
Traditional AEDC, which aims to identify and recognise various acoustic events, offers an additional modality to existing ADAS by exploiting acoustic signals for environmental monitoring. For instance, Horn signals are a type of acoustic warning for approaching vehicles, emergency services, railway crossings, and other hazards in urban areas. Integration of AEDC with vision-based ADASs addresses several complementary aspects: (1) auditory signals provide omnidirectional coverage; (2) horn signals encode semantic information about threat urgency and source

* Corresponding author.
E-mail addresses: up201912327@edu.fe.up.pt (V. Hajihashemi), up202003516@edu.fe.up.pt (A.A. Gharahbagh), jjmm@fe.up.pt (J.J.M. Machado), tavares@fe.up.pt (J.M.R.S. Tavares).

https://doi.org/10.1016/j.eswa.2026.133203
Received 14 January 2026; Received in revised form 2 June 2026; Accepted 7 June 2026
Available online 13 June 2026
0957-4174/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

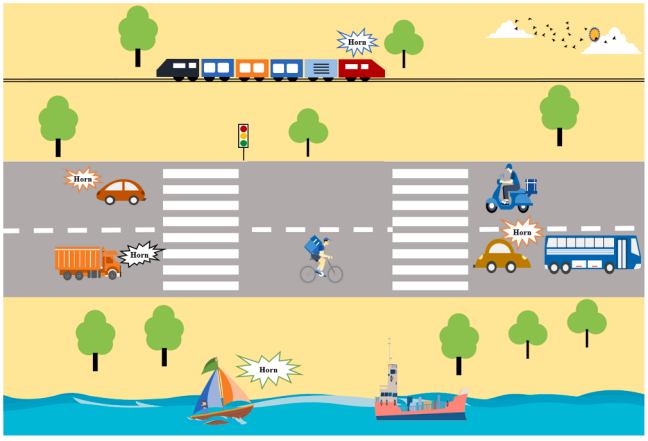


Fig. 1. Example of an urban environment and of sources that cause horn signals.

type unavailable in visual data; (3) acoustic processing has minimal computational overhead compared to video analysis; (4) sound propagation maintains effectiveness under visual degradation scenarios. An AEDC system typically comprises three stages: audio signal processing, feature extraction, and a machine learning classifier to detect the source of danger and, if necessary, alert the driver. Acoustic warning systems face unique challenges that distinguish them from existing AEDC systems. AEDC design must address several challenges: overlapping acoustic events in the environment, fast-response requirements, and accurate alerting for hazardous events. The above challenges and limited studies on detecting acoustic events in driving scenarios motivate the development of a hierarchical AEDC framework to detect horn signals and identify their type (Fig. 1). The main innovations made in the proposed AEDC method are:

1. Using shortened features in AEDC to reduce classifier complexity while maintaining detection accuracy;
2. Using transfer learning to accelerate learning and improve performance;
3. Analysing the effect of reducing input dimension on system performance;
4. Adding a stochastic-based post-processing step to enhance the AEDC system efficiency.

The remainder of this article is organised as follows: Section 2 reviews relevant AEDC literature, Section 3 details the proposed methodology, Section 4 presents experimental configuration and evaluation metrics, Section 5 analyses performance outcomes, and Section 6 provides concluding remarks with future research directions.

2. Literature review

To date, research on driving safety AEDC systems is limited. Several studies have employed acoustic signals to enhance driving safety (Chen & Thomas, 2020; Dewing, 2022; Du & Ding, 2014). An AEDC deep learning framework for driving safety was developed, focusing on two architectures: Convolutional Recurrent Neural Network (CRNN) and YOLOv5 (Castorena et al., 2024a; Jocher, 2020). For feature extraction, a Mel spectrogram was employed, and Mel frequency bands were optimised by reducing dimensions through pooling layers in a CRNN architecture (Castorena et al., 2024a). Driver auditory distractions were addressed using AEDC with Edge computing (Castorena et al., 2025). Because the primary task of all AEDC systems is to detect the types and timestamps of acoustic events, such as horn signals, these systems can be used to determine the types and occurrence times of the Horn in driving scenarios. In the current study, two types of Acoustic Event Detection (AED) systems were analysed. These include AED systems with and without feature selection. The first group provides an overview of feature selection and optimisation methods in AED systems, while the second group examines usage patterns and characteristics of Mel features in AEDC systems.

2.1. AED systems including feature selection

An Audio Event Classification (AEC) system combining a Mel spectrogram with a Support Vector Machine (SVM) classifier was presented in Ludena-Choez and Gallardo-Antolin (2016). To select the optimal Mel bands, a Mutual Information-based feature selection algorithm was employed. An AED system for classifying urban sounds was presented that employed Mel spectrogram features optimised by Class-conditional Dictionary Learning and a Probabilistic SVM (Ye et al., 2017). A combined framework was presented that includes Gammatone Filter Bank Energy features, followed by an encoder based on Rectified Linear Unit (ReLU) layers and frequency band optimisation (Mondal & Barman, 2022). An AED method utilising a feature selection scheme based on spectrogram features, determined by cross-correlation, was proposed in Hajihashemi et al. (2023). A framework for sound event detection based on Empirical Mode Decomposition and Intrinsic Mode Functions was proposed in Hajihashemi et al. (2025a), reducing input feature dimensions by more than 99% compared to state-of-the-art methods while maintaining detection accuracy. The same authors proposed a Feature-Reduction Scheme using a Two-Sample t-Test to remove Useless Spectrogram Bands in an AED system. A SELD method using a CRNN, which consists of Attention Feature Enhancement (AFE) and Parallel Multi-attention layers, was introduced in Chen and Huang (2024) to optimise feature representation. Multichannel features, adjacent spatial features, and a Conformer Block were employed to optimise Mel bands in an AED system (Dao et al., 2024). An AED system using an optimised set of Mel spectrogram, Mel-Frequency Cepstral Coefficients (MFCC), delta MFCC, and delta-delta MFCC features was presented in Mnasri et al. (2023). They employed a Variational Autoencoder (VAE) and Interval Type-2 Fuzzy Sets for classification. On the other hand, some studies employed simple preprocessing methods, such as cross-correlation and mutual information (Hajihashemi et al., 2023; Ludena-Choez & Gallardo-Antolin, 2016), for feature selection.

2.2. AED systems without feature selection

An MFCC AEC scheme that includes an SVM classifier was proposed in Hajihashemi et al. (2021). Statistical features were extracted from MFCC and applied to an SVM as a classifier (Jayalakshmi et al., 2018). An AEC method using a Gammatone spectrogram and a Linear SVM classifier was proposed in Mulimani and Koolagudi (2019). A deep learning approach based on Cochleogram features, derived from Gammatone filter banks, was proposed for AEDC in Hajihashemi et al. (2025b). A convolutional neural network (CNN) architecture with Mel spectrogram input was proposed in Kim et al. (2021). Adaptive Variational Modal Decomposition and the Pseudo Wigner-Ville Distribution were employed as feature extraction methods in conjunction with a CNN classifier for AED (Zhang et al., 2021). An AED that used a Mel spectrum as a feature and a CRNN classifier was presented in Zhao et al. (2022). A multitask learning (MTL) network based on a soft parameter-sharing mechanism with Log-Mel spectrogram input for AED was designed in Zhang et al. (2024). Another Acoustic Scene Classification (ASC) method suggested in Hajihashemi et al. (2022) includes two parallel ensemble classifiers as the initial step and a nonlinear mathematical fusion. Log-Mel-band energy was extracted for feature extraction, with CNN-Bidirectional Gated Recurrent Unit (CNN-BiGRU) and Transformer architectures employed for classification (Imoto et al., 2022).

An online Sound Event Localisation and Detection (SELD) system using a Mel spectrogram as the feature and a CRNN with ResNet Blocks as the classifier was proposed in Nagatomo et al. (2022). A drone sound detection system was developed by combining Log-Mel spectrogram and

MFCC features (Sun et al., 2024). A network for SELD that incorporates multi-head attention and depthwise separable convolution layers, utilising Log-Mel spectrogram features, was proposed in Zhou and Wan (2023). A joint framework for Audio Tagging and AED using a dual CNN was proposed in Liang et al. (2022). They used the Mel spectrogram as the main feature. The Log-Mel spectrogram was employed as input to the MSFF-Net architecture for AED (Wang et al., 2022). A hybrid PBA, Attention Capsule Network (AttCapsNet), and Bidirectional Gated Recurrent Unit (BGRU) for AED was presented in Meng et al. (2022). This model comprised Mel spectrogram features, an AttCapsNet for separating overlapping events, and a BGRU module. The HAAC (Hierarchical Audio Augmentation Chain) method for SELD was presented in Wu et al. (2023). They employed a Mel spectrogram for feature extraction and a CRNN and Event-Independent Network with multichannel processing capability (EINV2) for classification.

A Self-Supervised Learning method for AED was proposed in Wei et al. (2023). They employed a Mel spectrogram for feature extraction and Resnet18, VGG11, and a CRNN for classification. A hybrid Capsule-Transformer model for AED was introduced in Li et al. (2024). This model uses the Log-Mel spectrogram for feature extraction. A regression-based method that employed YOLO's one-dimensional bounding boxes and the Log-Mel spectrogram feature was presented in Liu et al. (2023). A method for AED combining a multiscale Dense CNN with Attention and RNN components was introduced in Yang et al. (2023). The method uses the Log-Mel spectrogram as the feature. An AED method using Log-Mel spectrogram and ResNet-18 was proposed in Hebbar et al. (2023). An AED method based on a Frequency-wise Transformer Encoder and a Local GRU decoder, incorporating nearest-neighbour interpolation and a BGRU, was proposed in Li et al. (2023). This study uses the Log-Mel spectrogram as the feature.

A Mel spectrogram, a pre-trained PaSST, a Vision Transformer trained on AudioSet, and openL3 embeddings, as well as a CNN trained with the audio-visual correspondence task, were used for AED (Angulo et al., 2023). An AED system using the Structural Causal Model and the back-door adjustment technique was proposed in Xin et al. (2023). The feature of this study was the Mel spectrogram. Log-Mel spectrogram features, Multi-dimensional Frequency Dynamic Convolution (MFDConv), and Confident Mean Teacher (CMT) networks were employed for AED (Xiao et al., 2023). An AED method employing Log-Mel spectrograms feature and a CRNN, where CNN blocks are replaced with dynamic multi-scale convolutional (DYMC) layers, was proposed in Wang et al. (2023a). The DCASE CRNN baseline system within a mean-teacher framework was used for AED (Serizel et al., 2023).

Their method employed Log-Mel Filterbank Energies as input. A semi-supervised AED model based on the pretrained audio neural network (PANN) was presented in Xu et al. (2023). PANNs performs feature extraction by combining the Wavegram and Log-Mel spectrogram. An AED system that consists of Log-Mel Energies as input and a CRNN classifier was introduced in Martín-Morató et al. (2023). Transfer learning for AED with Log-Mel spectrogram feature was employed in Luo et al. (2023). An Improved Self-consistency Training (ISCT) method with Log-Mel spectrogram input for the AED system was introduced in Wang et al. (2023b). A SELD method using Log-Mel spectrograms and a CNN with soft parameter sharing was presented in Yuan et al. (2023). The Event-Relation-Enhanced Regression (EERD)-SED algorithm for AED, utilising MFCC features and a Mobile-RegressionNet architecture for classification, was proposed in Tian et al. (2023). A real-time CNN system for AED, utilising Log-Mel spectrograms and MFCCs as input features, was developed in Chavdar et al. (2023).

A hybrid ResNet, Multi-Scale Channel Attention (MSCA) AED system with Log-Mel spectrogram features was introduced in Xue et al. (2023). A Unet was employed as an SELD system with Mel spectrogram features (Shin et al., 2023). An AED system using Log-Mel spectrograms with a CNN classifier was proposed in Cai et al. (2023). An AED system consisting of an optimised version of the Conformer network with a Log-Mel spectrogram feature was developed in Chan and Chin (2022). An AED system that uses MFCC features with a linear SVM was presented in Svatos and Holub (2023). A four-stage data augmentation framework was introduced to improve the performance of the ResNet Conformer model in SELD tasks, where Log-Mel spectrogram and Generalised Cross-Correlation Phase Transform (GCC-PHAT) are employed as main features (Wang et al., 2023c).

A hybrid algorithm based on CRNN for SELD in a 3D space was presented, which extracts input features by combining the Log-Mel spectrum and GCC-PHAT (Zhao et al., 2023). A semi-supervised framework for AED that employed Log-Mel spectrogram features and Frequency Dynamic (FDY) convolutions was presented in Khandelwal et al. (2023). Log-Mel spectrograms were processed by a shared CNN to maintain good temporal resolution as a feature (Gao et al., 2024). For the main classifier, they used a CRNN including a Conformer block. Most reviewed AEDC studies improve the separation of Mel frequency bands within deep learning layers (Angulo et al., 2023; Chen & Huang, 2024; Yang et al., 2023; Zhang et al., 2021), with no processing of frequency bands outside the network. Separating Mel frequency bands during DNN training is possible; however, increasing input dimensions raises DNN complexity and requires more training samples, thereby increasing computational complexity.

Despite the emergence of deep embeddings and end-to-end architectures, Mel and Log-Mel features remain dominant across AED and SELD systems (Hajihashemi et al., 2026). Several factors explain this persistence. First, these features are easy to interpret, allowing researchers to analyse frequency patterns directly and understand model decisions through clear time-frequency representations. Second, Mel-based features demonstrate robust performance across diverse acoustic environments, maintaining reliability even with limited training data and different environments. Third, computational efficiency remains a practical advantage, as Mel-spectrogram extraction requires significantly less processing power than deep embedding models, making real-time applications feasible on resource-constrained devices. However, a notable gap exists in the AEDC literature regarding horn detection and classification within road environments. While researchers have extensively studied general AED systems, few studies have addressed the challenges of separating and classifying horn signals under varying traffic conditions, engine noise, and environmental interference. Horn signals exhibit distinct temporal and spectral characteristics that differ from those of typical acoustic events. However, targeted research addressing these specific requirements is still limited. This gap highlights the need for specialised approaches that account for the overlapping nature of road sounds and the critical timing requirements of driving safety applications. Using transfer learning and pre-trained networks solves some of the above challenges, though the features they produce may contain information that is not relevant to the specific target event.

The proposed method addresses this limitation by applying mRMR feature selection after VGGish extraction, removing redundant or irrelevant features, and retaining only those most useful for separating horn types. This approach increases AEDC performance by preventing noise and unnecessary inputs from reaching the classifier. Table 1 shows a summarised list of state-of-the-art AED systems, including feature extraction method, classifier, and their pros and cons, which shows that most AEDC, AED, ASC, and SELD studies employ Mel and Log-Mel spectrograms as the primary feature. More recently, Transformer models have also gained attention in acoustic event classification. The Audio Spectrogram Transformer (AST) is a purely attention-based model built on the Vision Transformer (ViT) architecture with 87 million parameters and 12 encoder layers (Gong et al., 2021). AST achieved 0.485 mAP on AudioSet and 95.6% accuracy on ESC-50, and it remains one of the strongest models for general-purpose acoustic classification.

However, AST and similar Transformer architectures rely on large-scale pretraining datasets such as ImageNet and AudioSet, and they require GPU acceleration for both training and inference. These properties make them less suited to the embedded, real-time processing required by automotive safety applications. The proposed framework in this study takes

Table 1
Summary of AEDC systems based on features, classifiers, and their pros and cons (Part I).

Reference	Feature	Feature Reduction	Classifier	Advantages	Disadvantages
Kim et al. (2021)	Mel spectrogram	No	CNN	Accuracy	Not generalisable
Angulo et al. (2023)	Mel spectrogram	No	OpenL3 embeddings	Independent normalisation of each Mel band in the feature extraction step	Computational complexity
Cai et al. (2023)	Log-Mel spectrogram	No	CNN	Using Class Activation Map	Computational complexity
Castorena et al. (2024a)	Mel spectrogram	No	CRNN	Accuracy	Computational complexity
Castorena et al. (2025)	Mel spectrogram	No	CRNN and YOLOv5	Accuracy	Computational complexity
Chavdar et al. (2023)	Log-Mel spectrogram, MFCC	No	CNN	Real-time	Computational complexity
Hajihashemi et al. (2023)	Mel spectrogram	Cross-correlation	CNN	Using mathematical and statistical modules	Not generalisable
Chen and Huang (2024)	Log-Mel spectrogram	No	CRNN	Using parallel multi-attention (PMA) modules	Computational complexity
Dao et al. (2024)	Log-Mel spectrogram	No	CRNN + Conformer	Multi-channel and adjacent spatial features	Computational complexity
Gao et al. (2024)	Log-Mel spectrogram	No	LTE-Net	Optimising local temporal embedding	Computational complexity
Hajihashemi et al. (2021)	MFCC	No	SVM	Simple structure	Not generalisable
Jayalakshmi et al. (2018)	MFCC	No	SVM	Extra statistical features	Complexity
Khandelwal et al. (2023)	Log-Mel spectrogram	No	PANN	FDY-Conv were used to optimise spectral bands	Not generalisable
Li et al. (2024)	Log-Mel spectrogram	No	Capsule Network	Dynamic Routing algorithm along with Temporal Attention	Not generalisable
Ludena-Choez and Gallardo-Antolin (2016)	Mel spectrogram	Mutual Information	SVM	Explains suitable hardware for affordable Sound Event Detection (SED) in vehicles	Low accuracy
Mulimani and Koolagudi (2019)	Gammatone spectrogram	No	SVM	Simple structure	Not generalisable
Shin et al. (2023)	Mel spectrogram	No	SELDnet	Using U-Net for noise reduction	Computational complexity
Sun et al. (2024)	Log-Mel spectrogram, MFCC	No	CNN	Combining beamforming algorithms with a CNN	Computational complexity
(Svatos & Holub, 2023)	MFCC	No	SVM	Using the physical properties of an event	Not generalisable
Tian et al. (2023)	MFCC	No	Mobile-RegressionNet	Transfer learning	Not generalisable
Wang et al. (2023c)	Log-Mel spectrogram	No	ResNet-Conformer	Four-stage data augmentation	Computational complexity
Yang et al. (2023)	Log-Mel spectrogram	No	MS-AttDenseNet-RNN	Using multiscale dense CRNN with lightweight attention	Computational complexity
Ye et al. (2017)	Mel spectrogram	No	SVM	Processing features using normalisation and whitening	Not generalisable
Zhang et al. (2021)	Wigner-Ville Distribution	No	CNN	Mathematical structure	Computational complexity
Zhao et al. (2022)	Mel spectrogram	No	CRNN	Hierarchical structure	Computational complexity
Zhao et al. (2023)	Log-Mel spectrogram	No	CRNN	Channel-attention to enhance useful features and suppress noise	Not generalisable

a different path by combining pretrained feature extractors (VGGish and YAMNet) with lightweight traditional classifiers (SVMs), keeping computational load low enough for deployment on processors with limited resources. A detailed computational comparison between the proposed method and Transformer-based alternatives is presented in Section 5.10.

3. Proposed method

The AEDC method addressed in the current study comprises a two-step hierarchical scheme. Fig. 2 illustrates the analysed workflows for the AED step, where three schemes extract features from raw acoustic signals: Log-Mel spectrogram, VGGish, and YAMNet. Each feature representation feeds into two separate binary classifiers that discriminate the horn signal from background noise. Feature extraction in all three schemes is based on the Log-Mel spectrogram because the VGGish and YAMNet also use it in the first stage.

After selecting the best AED workflow, the detected horn segments are applied to the AEC step. Fig. 3 illustrates the four feature extraction schemes evaluated for the AEC step. These four schemes are not used together during operation; instead, they are tested separately to determine which yields the best classification results.

The four schemes are: (1) Log-Mel spectrogram, where the raw spectrogram is flattened into a one-dimensional vector; (2) VGGish, which uses its 128-dimensional embedding output; (3) YAMNet, which uses its 521-dimensional embedding output; and (4) Combined, which concatenates the VGGish, YAMNet, and Log-Mel vectors into a single feature vector of 6793 dimensions. Although VGGish and YAMNet both compute a Log-Mel spectrogram as their first internal processing stage, the standalone Log-Mel path differs in that it feeds the raw spectrogram values directly to the classifier, bypassing a deep network. Within each scheme, the feature vector first undergoes mRMR feature selection to reduce the dimensionality, and then passes through four parallel binary SVM classifiers, one for each horn class: boat, car, train, and truck.

These four classifiers run in parallel so that the system can detect multiple horn types at the same time from a single input frame. Post-processing modules based on a Binomial Probability Distribution Function (BPDF) operate on both AED and AEC outputs to correct errors across consecutive frames. Because horn signals last longer than a single analysis window, the same event is expected to appear in several consecutive frames. The BPDF block applies a majority voting rule: a

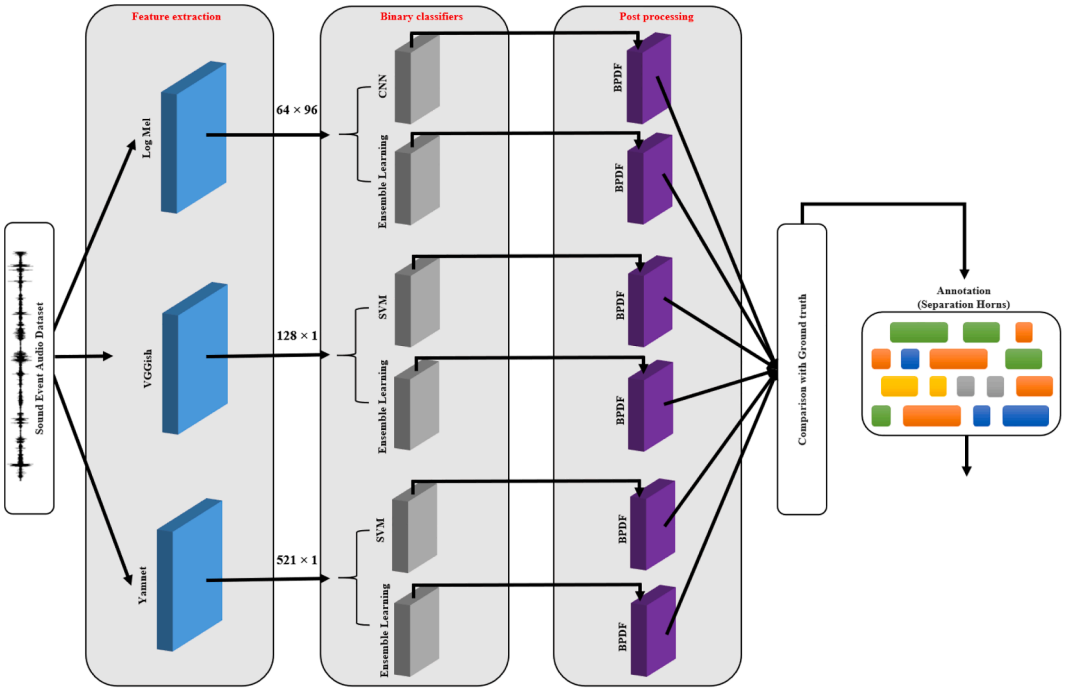


Fig. 2. The AED step of the proposed method for selecting the best AED system.

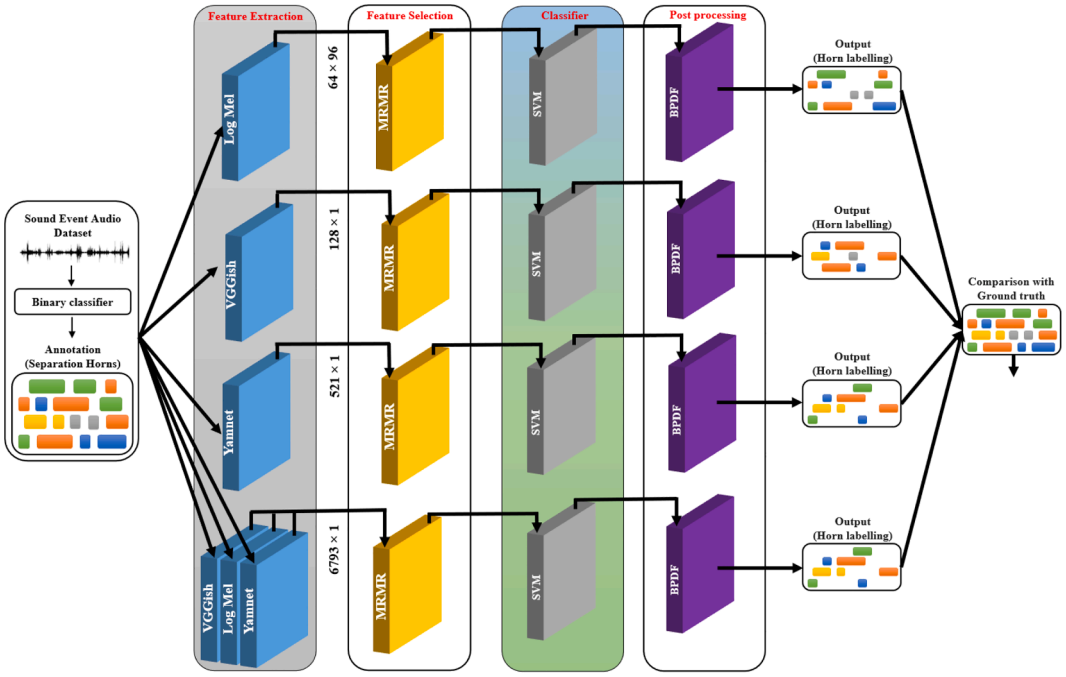


Fig. 3. The AEC step of the proposed method.

detection is accepted only if it occurs in at least 5 of the 10 consecutive frames. If fewer than five frames agree, the detection is treated as a false alarm and removed.

In the AED step, the BPDF block outputs a corrected binary label (horn or no horn), which is then passed to the AEC step. In the AEC step, each of the four parallel classifiers has its own BPDF block, and the corrected output is a binary label for each horn class (boat, car, train, or truck). The comparison block (Figs. 2 and 3) compares these corrected labels against the ground truth annotations to compute the evaluation metrics reported in Section 5. This hierarchical separation enables the independent optimisation of detection sensitivity and classification accuracy, while reducing computational complexity compared to end-to-end deep learning approaches. This Section presents the details of the Log-Mel spectrogram, VGGish, YAMnet, and mRMR feature selection approaches.

3.1. Log-Mel spectrogram

The Mel scale maps linear frequencies to a perceptual scale that reflects how humans hear sound, placing finer resolution at lower fre-

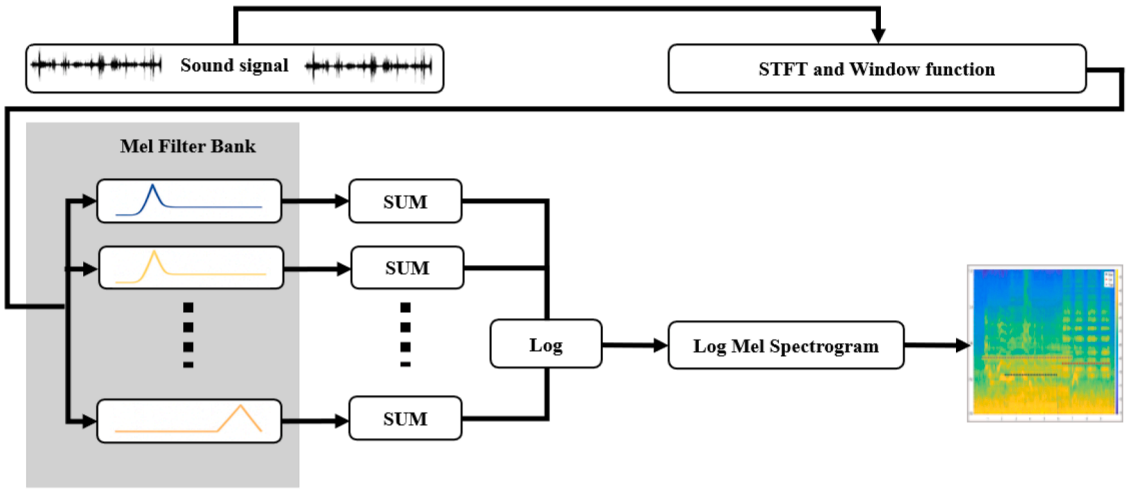


Fig. 4. Block diagram of the Log-Mel spectrogram.

quencies and coarser resolution at higher ones (Mesaros et al., 2017). The conversion from Hz to the Mel scale follows:

$$f_{\text{Mel}} = 2595 \cdot \log_{10}\left(1 + \frac{f_{\text{Hz}}}{700}\right), \tag{1}$$

where f_{Hz} is the frequency in Hz and f_{Mel} is the corresponding frequency on the Mel scale. To build the Mel spectrogram, the raw audio is first split into overlapping frames. Each frame is then transformed into the frequency domain through the Short-Time Fourier Transform (STFT) (Mesaros et al., 2017). The resulting spectrum is passed through a bank of k triangular filters equally spaced on the Mel scale. Each filter H_k weights the squared STFT magnitudes, and the summed energy per band gives the Mel spectrogram:

$$X_{\text{Mel}}(k) = \sum_{m=0}^{N-1} \left[|X(m)|^2 \, H_k(m)\right], \tag{2}$$

where $X(m)$ is the STFT magnitude spectrum, m is the frequency bin index, N is the FFT length, k is the filter index, and $H_k(m)$ is the kth triangular Mel filter weight. Applying a logarithmic compression to $X_{\text{Mel}}(k)$ produces the Log-Mel spectrogram used throughout this study (Diwakar & Gupta, 2024; El-Latif et al., 2024; Purwins et al., 2019). Fig. 4 shows the complete workflow. Because VGGish and YAMNet both use the Log-Mel spectrogram as their first processing stage, this feature serves as a common foundation across all three extraction paths in the proposed framework.

3.2. VGGish

VGGish is a deep CNN based on the VGG-E architecture, originally designed for image classification but widely adopted for acoustic feature extraction (Diwakar & Gupta, 2024; El-Latif et al., 2024; Hershey et al., 2017). The model was trained on Google's YouTube-100M (100 million videos) and AudioSet (over 5.24 million hours of labelled audio) datasets, which gives it strong tolerance to noise and makes it well-suited for environments like urban roads and highways (El-Latif et al., 2024; Hershey et al., 2017). Fig. 5 shows the block diagram of VGGish. The architecture consists of 13 convolutional layers with 3×3 filters, taking a Log-Mel spectrogram as input. The input parameters used in this study are as follows:

- Time frame length = 25 ms;
- Overlap length = 15 ms;
- Total signal length = 0.975 s;
- Number of Mel filters = 64;
- Window function: Hanning;
- Frequency range [125 7500] Hz (Mel filter bank coverage);
- Input dimension = 64×96;
- Output dimension = 1×128.

VGGish produces a 128-dimensional feature vector (embedding) for each 0.975 s acoustic frame. The current study uses this embedding as the input feature vector for the AEC step. Because VGGish was pre-trained on a very large and diverse corpus, the learned representations capture acoustic patterns that go well beyond what task-specific training on a smaller dataset could achieve, which is particularly useful for separating horn types in noisy driving conditions.

3.3. YAMNet network

YAMNet is a CNN built on the MobileNetV1 architecture, developed by Google for SED (Howard, 2017). It uses depthwise separable convolutions to maintain a good balance between accuracy and computational cost, and it can recognise 521 different acoustic events after training on the AudioSet dataset (Avila & Pinzón, 2024; Malmberg, 2021). The YAMNet architecture is depicted in Fig. 6. The network contains 27 convolutional layers followed by global average pooling and a fully connected layer, with only 3.7 million parameters compared to 72.1 million in VGGish (Avila & Pinzón, 2024; Hershey et al., 2017).
This lighter structure makes it a practical choice for edge devices and real-time automotive applications. YAMNet takes the same Log-Mel spectrogram input as VGGish (resampled to 16 kHz mono, with identical parameters listed in Section 3.2) and produces a 521-dimensional embedding vector from its final softmax layer. The current study uses this 1×521 vector as the input feature for the AEC step, allowing a direct comparison with the 128-dimensional VGGish embedding under the same mRMR selection and SVM classification pipeline.

3.4. CNN classifier

In the AED step (Fig. 2), a simple two-dimensional (2D) CNN with Log-Mel spectrogram input is one of the classifiers employed to detect horn occurrence. Fig. 7 shows the structure of the employed classifier. Algorithm 1 describes the operation of the implemented CNN.

3.5. Ensemble classifiers

In the AED step (Fig. 2), seven ensemble learning methods were evaluated alongside the CNN and SVM classifiers to identify the best approach for separating horn signals from background noise. Ensemble methods train multiple weak learners and combine their outputs to

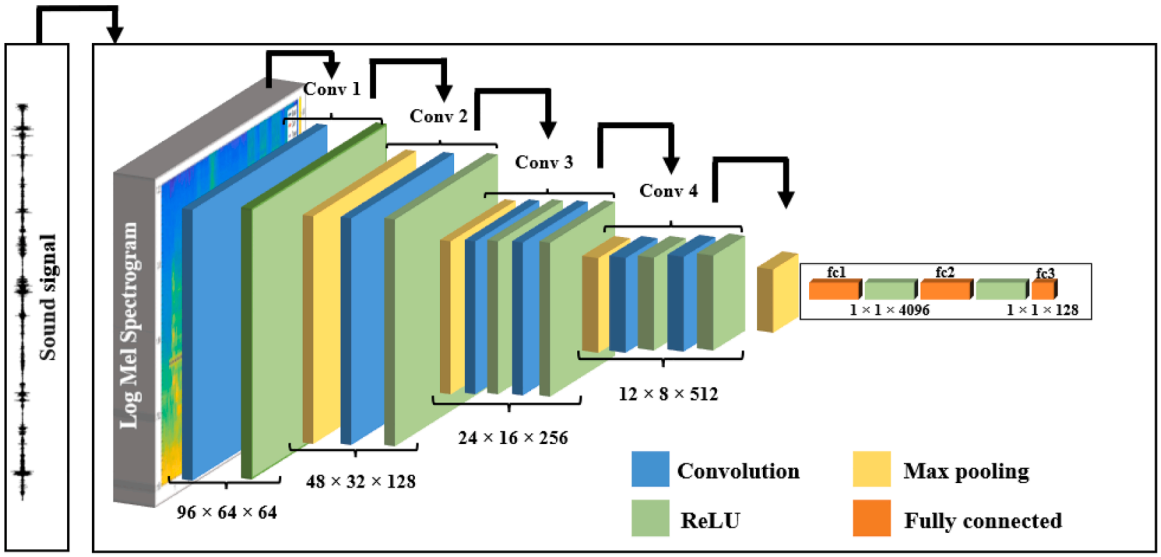


Fig. 5. The VGGish architecture.

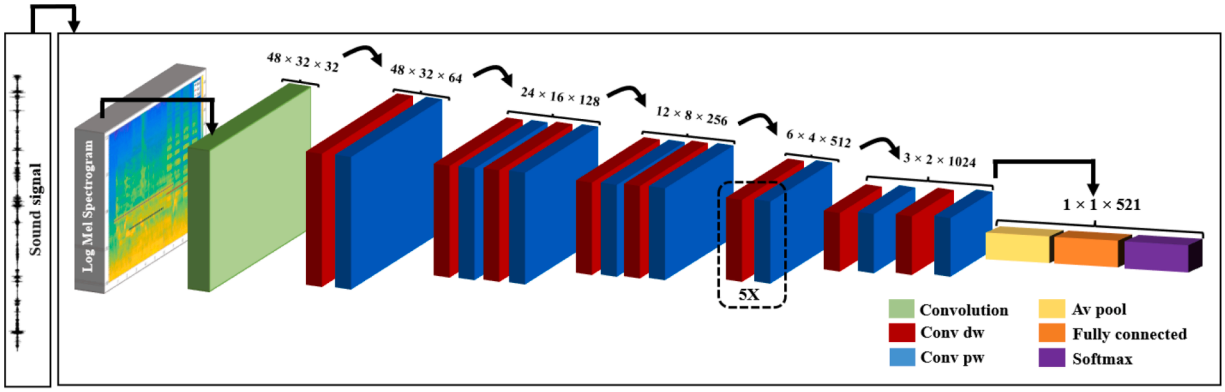


Fig. 6. The YAMNet architecture.

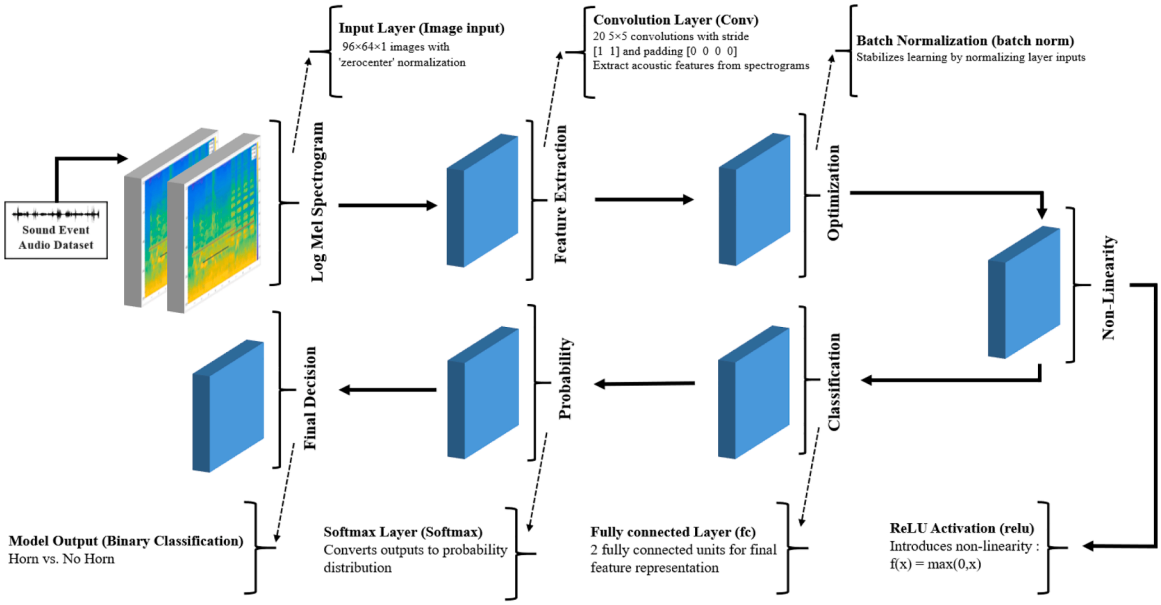


Fig. 7. Structure of the CNN binary classifier employed in the AED step.

Algorithm 1: CNN architecture for Horn detection using Log-Mel spectrogram.

Input: Input spectrogram $X \in \mathbb{R}^{96\times64\times1}$
Output: Class probabilities $P \in \mathbb{R}^2$

$X_{norm} \leftarrow \text{ZeroCenter}(X)$; // Image input with zero-center normalisation
$F_1 \leftarrow \text{Conv2D}(X_{norm}, \text{filters} = 20, \text{kernel_size} = 5 \times 5,$ $\text{stride} = [1,1], \text{padding} = [0,0,0,0])$; // 2D Convolution
$F_2 \leftarrow \text{BatchNorm}(F_1)$; // Batch normalisation
$F_3 \leftarrow \text{ReLU}(F_2)$; // ReLU activation
$F_4 \leftarrow \text{Flatten}(F_3)$; // Flatten for fully connected layer
$F_5 \leftarrow \text{FC}(F_4, \text{units} = 2)$; // Fully connected layer
$P \leftarrow \text{Softmax}(F_5)$; // Output probabilities

return P

produce a stronger overall prediction. The seven methods tested in this study are:

1. **Bag (Bootstrap Aggregation):** trains each learner on a random bootstrap sample of the training data and averages their predictions. This reduces variance and tends to produce stable results across different runs.
2. **AdaBoostM1:** assigns higher weights to misclassified samples in each round so that later learners focus on the harder cases.
3. **LogitBoost:** fits additive logistic regression models in a stage-wise manner, which makes it less sensitive to noisy samples than AdaBoostM1.
4. **GentleBoost:** a variant of AdaBoost that uses a gentler weight update rule and is more resistant to outliers.
5. **RobustBoost:** designed to maintain accuracy even when the training data contains label noise or outliers.
6. **RUSBoost:** combines Random Under Sampling with boosting, which helps when classes are imbalanced by reducing the majority class before each boosting round.
7. **Subspace:** trains each learner on a random subset of features rather than a random subset of samples, which can help when the feature space is large.

All seven methods were tested with VGGish, YAMNet, and Log-Mel spectrogram features. The comparative results in Table 4 show that the Bag ensemble with Log-Mel spectrogram features produced the best AED performance (F1 = 0.79 on the test set). Section 5.3 provides the specific training parameters for each ensemble method.

3.6. mRMR feature selection

Some studies have proposed methods to reduce the dimensionality of Log-Mel spectrograms, VGGish, and YAMNet features in AED systems (Avila & Pinzón, 2024; Liu et al., 2025). The VGGish model's ability to extract subtle acoustic features was demonstrated in Hershey et al. (2017). They demonstrated that the Log-Mel spectrogram image distinctly separates different acoustic events. Based on this finding, selecting the right features from the Log-Mel spectrogram, VGGish, and YAMNet for each acoustic event can increase AEDC efficiency. In this context, the feature vectors from all three methods are not equally relevant to horn detection, and some may contain redundant or irrelevant information. This issue can increase computational costs during classification stages.

The mRMR method can solve this problem by selecting optimal features and reducing dimensions. The mRMR feature selection method minimises information overlap while ensuring each selected feature significantly predicts the output (Peng et al., 2005). By creating an optimal balance between relevance (connection to the output) and redundancy (reducing overlap), mRMR substantially improves system accuracy and efficiency. In this regard, the effectiveness of the mRMR method for high-dimensional data has been confirmed (Peng et al., 2005). In the mRMR method, relevance refers to the extent to which a feature influences the output prediction; in this study, the relevance is measured in terms of the horn type. Relevance is calculated using an index called mutual information, which quantifies the amount of information shared between a feature and the output. To calculate the relevance between a feature X_i and output Y, one can use:

$$I(X_i;Y) = \sum_{x\in X_i}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{3}$$

where the function $p(x,y)$ describes the joint probability of X_i and Y. These values are obtained from the training data. Similarly, $p(x)$ and $p(y)$ represent the marginal probabilities for X_i and Y, respectively, indicating their distributions in the dataset. The logarithm in this equation is typically defined with base 2, so its output quantitatively demonstrates how much knowing the value of X_i reduces uncertainty about Y. Higher values of $I(X_i;Y)$ mean that feature X_i has a good output predictor and will consequently have higher priority in the feature selection process. The concept of Redundancy evaluates how much information overlaps between a candidate feature and previously selected features. This criterion, confirmed by recent studies (Ding & Peng, 2005), demonstrates the extent to which a feature provides repetitive information compared to other existing features. This criterion is defined as the average mutual information between the candidate feature X_i and all other features, X_j:

$$I_{x_i} = \frac{1}{|N_s|}\sum_{j=1,\, i\neq j}^{N_s} I(X_i;X_j), \tag{4}$$

where N_s represents the total feature count, and $I(X_i;X_j)$ measures the mutual information between the candidate feature X_i and each feature X_j from the set of existing features. High values for $I(X_i;X_j)$ indicate significant information overlap between features, and the average of this value expresses how a feature is similar to all existing features. By minimising this criterion, the mRMR algorithm ensures that the final features are informationally diverse. In the mRMR method, the feature selection criterion is designed to simultaneously combine two measures: relevance and redundancy, into a single objective function. The objective function is defined as:

$$\phi(X_i) = I(X_i;Y) - I_{x_i}, \tag{5}$$

where $\phi(X_i)$ is the final score of X_i, and features with a higher $\phi(X_i)$ will be more critical. Before calculating the final score, the feature values were divided into different intervals for discretisation. The number of intervals (bins), k, is a crucial parameter in the discretisation process. If k is too small, it can lead to oversimplification and loss of essential features. On the other hand, if k is too large, the noise effect on probability estimates increases. After discretisation, probability distributions such as $p(x,y)$, $p(x)$, and $p(y)$ are extracted and used to calculate relevance and redundancy. Although discretisation is common, alternatives like kernel density estimation can provide discrete probability distribution estimates without requiring division into bins, although these methods have higher computational costs. Adaptive partitioning has been suggested to improve discretisation in continuous data (Darbellay & Vajda, 1999).

The mRMR method finds the optimal solution using a greedy forward selection algorithm. The effectiveness of greedy approaches in finding optimal solutions in information theoretic contexts has been confirmed by Battiti (1994). In greedy forward selection, the algorithm initially assumes that the subset of selected features S is empty and selects the feature with the highest mutual information $I(X_i;Y)$ value. At this stage, the information overlaps within the solution set, or redundancy equals zero. In the next iteration, a feature which maximises the objective function $\phi(X_i)$ is added to S. I_{x_i} is calculated only with respect to the selected features S. This process continues until all features are added to S, which creates a sorted list of features. This ongoing update ensures that feature selection remains optimal in terms of both predictive power and

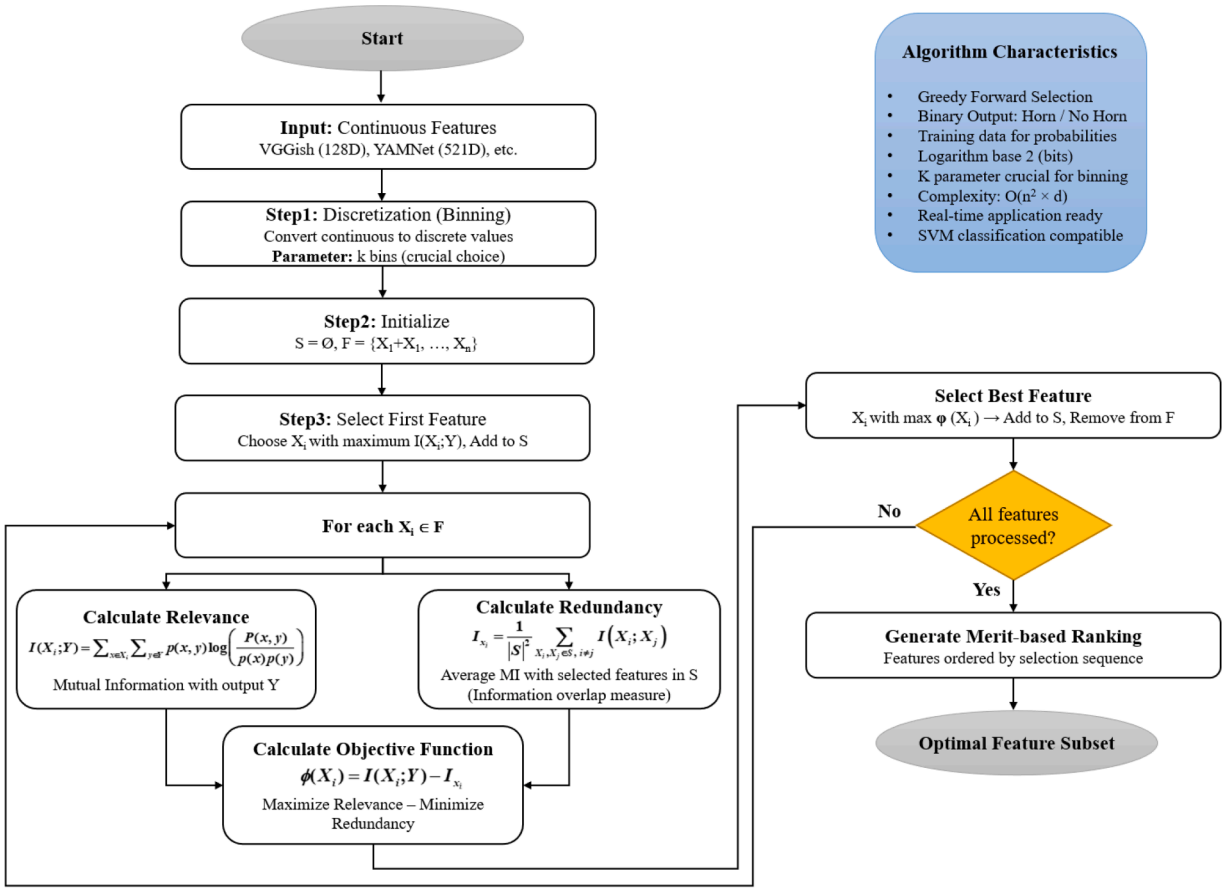


Fig. 8. Flowchart of the mRMR method.

information diversity. Fig. 8 depicts the flowchart of mRMR. Although this method is not ideal for reaching globally optimal solutions, it is very efficient because each iteration evaluates only the remaining features of $|S| - n$. This characteristic enables feature selection in high-dimensional data, especially in real-time applications. The sorted feature list enables the assessment of how increasing the feature count affects the performance of the SVM classifier in the proposed AEC system, allowing for the selection of the optimal feature count by balancing computational cost and performance.

This selection yields a streamlined feature subset that enables faster and more stable SVM training, which is crucial for real-time applications in driving scenarios. To summarise the complete pipeline: a 10-second audio clip is split into 1-second frames with a 0.1-second hop. In the AED step, a Log-Mel spectrogram (64 bands) is extracted from each frame and classified as horn or no horn by a Bag ensemble. The BPDF block then corrects the frame-level outputs using majority voting over ten consecutive frames. Frames labelled as horn are forwarded to the AEC step, where VGGish extracts a 128-dimensional embedding from each frame. The mRMR algorithm ranks all 128 features, and the top 80 are selected. Four parallel binary SVM classifiers (RBF kernel), one per horn class, receive the reduced feature vector and produce independent predictions. A second BPDF block corrects each classifier output. The full parameter settings for every component are given in Section 5.3.

4. Experimental setup

This Section discusses the details of the implementation and evaluation metrics of the proposed method. Firstly, the Section 4.1 addresses the database, followed by evaluation criteria in Section 4.2. Finally, Section 4.3 describes the implementation details, parameters, and evaluation metrics.

4.1. Database

This study uses a database specifically designed for AEDC systems in driving scenarios (Castorena et al., 2024a). Since the primary goal of the proposed horn detection system is to enhance driving safety, the chosen database aligns perfectly with the research objective. This comprehensive database includes acoustic events that can distract drivers in real conditions and various driving scenarios. The data was synthetically generated using real data and noise from driving environments and inside vehicles, along with changes in amplitude and characteristics of the original acoustic signal. The audio clips in this database are all 10 seconds long and were selected from various sources, including UrbanSound (Salamon et al., 2014), Musical genre classification of acoustic signals (Tzanetakis & Cook, 2002), Audio Set (Gemmeke et al., 2017), Old phone ringtones as midi (Torosyan, 2019), ASR-CabNois (MagicHub, 2023), Freesound (Group, 2023), A singing voice (Wilkins et al., 2018), DCASE 2020 Task 4 dataset (Serizel et al., 2020). A cry corpus features dataset (Valani, 2023).
Specifically, the horn signals were taken from references (Group, 2023; Salamon et al., 2014). Scaper, as a Python library, was used to combine event acoustic signals with environmental sounds/noise in the database (Castorena et al., 2024a; Valani, 2023). In the AEDC system proposed in Castorena et al. (2024a), acoustic events are categorised into two main groups: those inside and outside the car, with each group having its own subgroups. Fig. 9 depicts a summarised taxonomy of this database, including horn signals. The database contains 46 distinct acoustic event categories. The database contains 14,952 original 10-second audio files

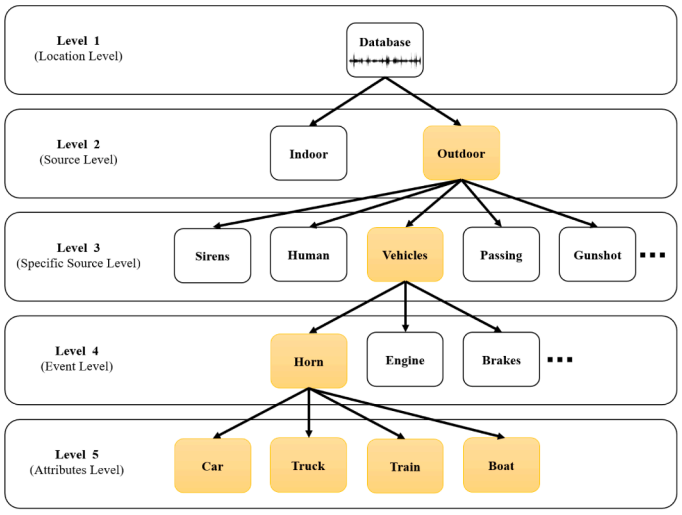


Fig. 9. Taxonomy of the used database, including selected classes.

for training, 1996 for testing, and 1991 for validation. After segmentation using 1-second windows with 0.1-second hop length, this yields the sample counts shown in Table 2.

4.2. Evaluation criteria

To evaluate the proposed method, Polyphonic SED (Mesaros et al., 2016) evaluation metrics were employed. Segment-based evaluation indicators were used since the database contains multiple simultaneous horn events. In this standard, acoustic events are detected based on small audio segments. As mentioned, the default segment length in YAMNet and VGGish is 0.975 seconds, and this same length was applied to Mel in the first step. It is assumed that if an event exists for more than 0.1 seconds, it exists in that segment. The following definitions and metrics were used in the evaluation:

- TP: Number of times the model correctly detected the presence of an event in the segment;
- FP: Number of times the model detected an event in the segment when no event had occurred;
- FN: Number of times the model output indicated no event in the segment when an event had occurred;
- TN: Number of times the model correctly detected the absence of an event in the segment.

Precision (Pr) denotes the ratio of correctly detected segments containing acoustic events to the total number of segments containing acoustic events:

$$\mathrm{Pr} = \frac{TP}{TP + FP}. \tag{6}$$

Recall (Re) denotes the ratio of correctly detected segments containing acoustic events to the total number of segments the system identified as containing acoustic events:

$$\mathrm{Re} = \frac{TP}{TP + FN}. \tag{7}$$

The F score (F1) represents the ratio of correctly detected segments containing acoustic events to the sum of correctly detected segments containing acoustic events and half of the total system errors:

$$F1 = 2\frac{\mathrm{Recall} \times \mathit{Precision}}{\mathrm{Recall} + \mathit{Precision}} = \frac{TP}{TP + \frac{(FP+FN)}{2}}. \tag{8}$$

Accuracy (Acc) represents the ratio of all correctly classified segments to the total number of segments:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{9}$$

Table 2
Number of samples with and without horns and each horn class, in the used database (Castorena et al., 2024a).

Split	No Horn	Horn	Boat	Car	Truck	Train
Train	1,064,832	295,800	105,762	55,866	64,179	91,934
Test	143,018	38,618	6273	6451	16,268	12,195
Validation	140,167	41,014	13,594	8240	8350	13,796

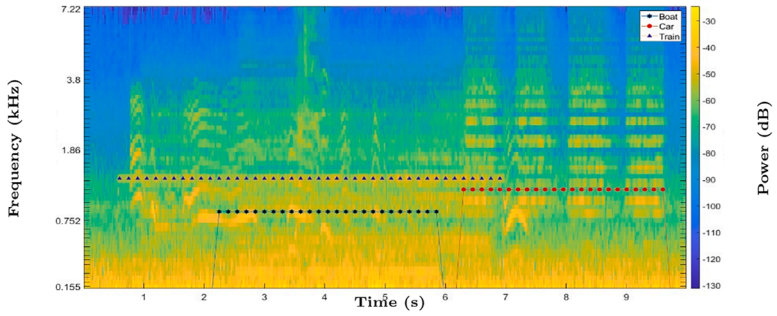
Table 3
Number of samples after data augmentation based on a 1-second input window.

Split	No Horn	Horn	Boat	Car	Truck	Train
Train	1,064,832	1,064,832	380,726	201,109	231,034	330,947
Test	143,018	143,018	23,231	23,891	60,247	45,163
Validation	140,167	140,167	46,458	28,161	28,536	47,148

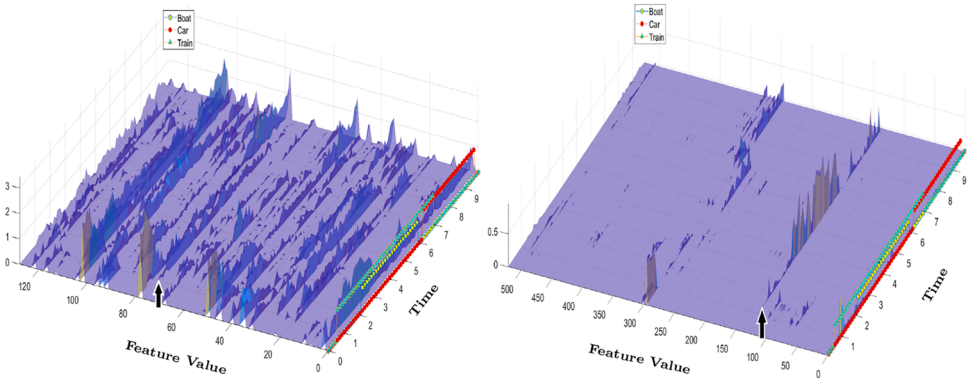
4.3. Experimental conditions

The evaluation approach performed in this study for the AED and AEC steps is a comparative method. For the AED binary classifier that separates horn signals from others, six approaches were analysed, including three feature extraction methods and three binary classifiers: SVM, CNN, and Ensemble learning (see Fig. 2). There was no feature selection in the AED step, and the standard parameters of the approaches were identical in both training and testing. An analysis of Mel parameters was conducted for the Log-Mel spectrogram, and various comparisons were performed to select the best classifier parameters. The comparative AEC step includes feature extraction methods based on the Log-Mel spectrogram, the VGGish network, the YAMNet network, and a combination of all three features together (see Fig. 3). Similar to the AED step, identical training and test data were employed in all comparisons to evaluate the approaches under similar conditions.
The SVM classifier and feature selection parameters were considered identical in all methods to avoid any potential bias in the results. Since VGGish and YAMNet networks share identical Mel feature extraction parameters, the comparative approach designed in Fig. 2 and 3 employed the same Mel parameters as these two networks. In the binary classification stage, the effect of Mel parameters on the output was examined through systematic ablation studies reported in Table 6.
The simulations in Figs. 2 and 3 were performed with 1-second input data and a hop length of 0.1 seconds. The selected timing parameters result in 91 samples from each 10-second clip. Any input sample containing more than 10% (0.1 s) of the horn signal was considered a horn sample. Table 2 demonstrates the number of samples with and without horns and each horn type in the training, testing, and validation sets based on the selected frame and hop length. Table 2 demonstrates that the dataset is imbalanced in both AED and AEC steps.

In the case of AED, there is more than a threefold imbalance between the two classes, which must be considered in training. There is less imbalance among the horn classes. For example, the training group has approximately 105,700 boat samples compared to about 55,800 car samples, indicating a roughly twofold difference. The sample counts in Table 2 show a clear imbalance in both AED and AEC steps. In the AED step, non-horn samples outnumber horn samples by more than three to one, and within the horn classes, the boat class has roughly twice as many samples as the car class. To address this, the SCAPER library was used to generate additional labelled samples through pitch shifting and time stretching until each class reached a balanced count (Salamon et al., 2017). Table 3 shows the sample counts after augmentation. For the AED step, horn samples were increased from 295,800 to 1,064,832 to match the non-horn count. For the AEC step, each horn class was scaled in proportion to the training split so that the ratio between classes became roughly equal.



(a) Log-Mel spectrogram feature in a 10-second audio clip.



(b) VGGish embedding feature in a 10-second audio clip.

(c) YAMNet embedding feature in a 10-second audio clip.

Fig. 10. Feature representations of a 10-second audio clip with three overlapping horn events: (A) Log-Mel spectrogram, (B) VGGish embeddings, and (C) YAMNet embeddings.

Augmentation was carried out separately for the training, test, and validation sets, where each set was augmented using only its own data. This keeps the three partitions independent and prevents any information leakage between them. The augmentation ratio was kept equal across all classes so that each horn type received the same level of balancing. In some of the augmented clips, multiple horn signals appear at the same time, which reflects real driving conditions where horns from different sources can overlap. After applying data augmentation, the data in Table 2 changed to the one in Table 3.

5. Results

5.1. Features at a glance

In the first step, the three features were graphically plotted, and a visual analysis of their changes was performed during the horn. Fig. 10a shows the Log-Mel spectrogram of a 10-second clip containing three different horn signals. The horns also overlap. As can be seen from Fig. 10a, separating different horns from each other and the background noise is complex, and no distinct feature in the image can be considered representative of a particular horn event. Fig. 10b shows the VGGish embedding feature for the same clip. In this feature, relatively notable differences between the horns are visible. For example, when a train horn occurs, specific changes can be observed in feature values around index 80 in Fig. 10b (indicated by an arrow). The YAMNet feature is demonstrated in Fig. 10c. In this plot, some changes are visible. Still, no clear relationship can be concluded given that the horns existed between 0.5 and 9.6 seconds, and the pattern of changes is interrupted (for instance, the black arrow in Fig. 10c).

5.2. AED step

Based on Fig. 2, in the first step, horn signals were separated from non-horn signals using six binary classifiers trained with VGGish, YAMNet, and Mel features. In VGGish and YAMNet, the outputs are one-dimensional, whereas in the Log-Mel spectrogram, the output is two-dimensional. The specifications of the classifiers used are as follows: For classifiers with VGGish and YAMNet features, Support Vector Machine (SVM) and Ensemble learning structures with different kernels, including Radial Basis Function (RBF), polynomial, and linear, were used. For structures using Mel features, CNNs and Ensemble learning were used. In all features, the window length is assumed to be 0.975 seconds. In the CNN classifier (see Fig. 7), the Adam optimiser with an Initial Learning Rate of 0.01 yielded the best results, as reported. The results are presented in Table 4, where it is possible to perceive that the Log-Mel spectrogram feature in Bag-type Ensemble learning performs better than other methods.

To support claims of superiority in Bag-type Ensemble learning, the Analysis of Variance (ANOVA) test was used to assess the repeatability and reproducibility of the results. In the repeatability check, the classifiers were trained 10 times using the same train, test, and validation sets. In the reproducibility check, all classifiers were trained 10 times using new training, test, and validation sets randomly selected from the main set. In terms of repeatability and reproducibility of results, ANOVA confirmed the significance of differences between different classifiers ($F > 15.3, p < .001$) and found no significant differences between results within a given classifier ($p > .05$). All analyses were performed using a significance level of 0.05.

Fig. 11 shows the evaluation metrics for AED systems with the Log-Mel spectrogram feature. The Bag and RUSBoost ensemble's results are more

Table 4
The AED step evaluation results.

			Test								Validation							
			TN	FP	FN	TP	ACC	Pr	Re	F1	TN	FP	FN	TP	ACC	Pr	Re	F1
VGG	SVM	RBF	86,040	56,978	50,472	92,546	0.62	0.62	0.65	0.63	84,325	55,842	49,466	90,701	0.63	0.63	0.66	0.64
		polynomial	87,098	55,920	79,666	63,352	0.53	0.53	0.44	0.48	85,362	54,805	78,078	62,089	0.52	0.52	0.44	0.48
		linear	62,667	80,351	44,480	98,538	0.56	0.55	0.69	0.61	61,418	78,749	43,594	96,573	0.56	0.55	0.69	0.61
	Ens.	LogitBoost	80,854	62,164	54,057	88,961	0.59	0.59	0.62	0.60	79,242	60,925	52,980	87,187	0.59	0.58	0.63	0.60
		AdaboostM1	79,928	63,090	56,142	86,876	0.58	0.58	0.61	0.59	78,334	61,833	55,023	85,144	0.59	0.58	0.62	0.60
		Bag	124,042	18,976	91,086	51,932	0.62	0.73	0.36	0.49	121,569	18,598	89,270	50,897	0.61	0.71	0.36	0.48
		Subspace	57,207	85,811	38,561	104,457	0.57	0.55	0.73	0.63	56,066	84,101	37,792	102,375	0.57	0.56	0.74	0.63
		GentleBoost	106,252	36,766	79,636	63,382	0.59	0.63	0.44	0.52	104,134	36,033	78,048	62,119	0.60	0.65	0.45	0.53
		RobustBoost	116,687	26,331	98,266	44,752	0.56	0.63	0.31	0.42	114,361	25,806	96,307	43,860	0.57	0.64	0.31	0.42
		RUSBoost	74,998	68,020	54,379	88,639	0.57	0.57	0.62	0.59	73,503	66,664	53,295	86,872	0.57	0.56	0.63	0.60
YAMNet	SVM	RBF	84,375	58,643	69,251	73,767	0.55	0.56	0.52	0.54	82,693	57,474	67,870	72,297	0.54	0.55	0.51	0.53
		polynomial	83,784	59,234	68,711	74,307	0.55	0.56	0.52	0.54	82,114	58,053	67,341	72,826	0.55	0.56	0.51	0.53
		linear	87,123	55,895	72,468	70,550	0.55	0.56	0.49	0.52	85,387	54,780	71,024	69,143	0.55	0.56	0.50	0.53
	Ens.	LogitBoost	75,645	67,373	54,979	88,039	0.57	0.57	0.62	0.59	74,137	66,030	53,883	86,284	0.57	0.56	0.61	0.59
		AdaboostM1	75,461	67,557	56,076	86,942	0.57	0.56	0.61	0.58	73,957	66,210	54,958	85,209	0.57	0.57	0.61	0.59
		Bag	94,007	49,011	65,963	77,055	0.60	0.61	0.54	0.57	92,133	48,034	64,648	75,519	0.59	0.60	0.53	0.56
		Subspace	87,268	55,750	75,687	67,331	0.54	0.55	0.47	0.51	85,528	54,639	74,178	65,989	0.54	0.54	0.47	0.51
		GentleBoost	77,959	65,059	58,156	84,862	0.57	0.57	0.59	0.58	76,405	63,762	56,997	83,170	0.56	0.56	0.59	0.57
		RobustBoost	71,227	71,791	53,413	89,605	0.56	0.56	0.63	0.59	69,807	70,360	52,348	87,819	0.56	0.55	0.62	0.58
		RUSBoost	45,982	97,036	33,253	109,765	0.54	0.53	0.77	0.63	45,065	95,102	32,591	107,576	0.55	0.53	0.78	0.63
Log-Mel	CNN	-	117,519	25,499	41,271	101,747	0.77	0.80	0.71	0.75	115,177	24,990	40,448	99,719	0.77	0.80	0.72	0.76
	Ens.	LogitBoost	117,750	25,268	36,486	106,532	0.78	0.81	0.74	0.78	115,402	24,765	35,758	104,409	0.78	0.80	0.76	0.78
		AdaboostM1	117,869	25,149	38,485	104,533	0.78	0.81	0.73	0.77	115,520	24,647	37,718	102,449	0.78	0.80	0.75	0.77
		Bag	115,320	27,698	31,750	111,268	0.79	0.80	0.78	0.79	113,021	27,146	31,117	109,050	0.78	0.79	0.76	0.77
		Subspace	99,603	43,415	69,548	73,470	0.61	0.63	0.51	0.57	97,617	42,550	68,161	72,006	0.60	0.62	0.52	0.57
		GentleBoost	117,778	25,240	39,223	103,795	0.77	0.80	0.73	0.76	115,430	24,737	38,441	101,726	0.77	0.81	0.72	0.76
		RobustBoost	116,518	26,500	41,062	101,956	0.76	0.79	0.71	0.75	114,195	25,972	40,244	99,923	0.77	0.81	0.71	0.76
		RUSBoost	106,596	36,422	35,245	107,773	0.75	0.75	0.75	0.75	104,471	35,696	34,542	105,625	0.76	0.76	0.76	0.76

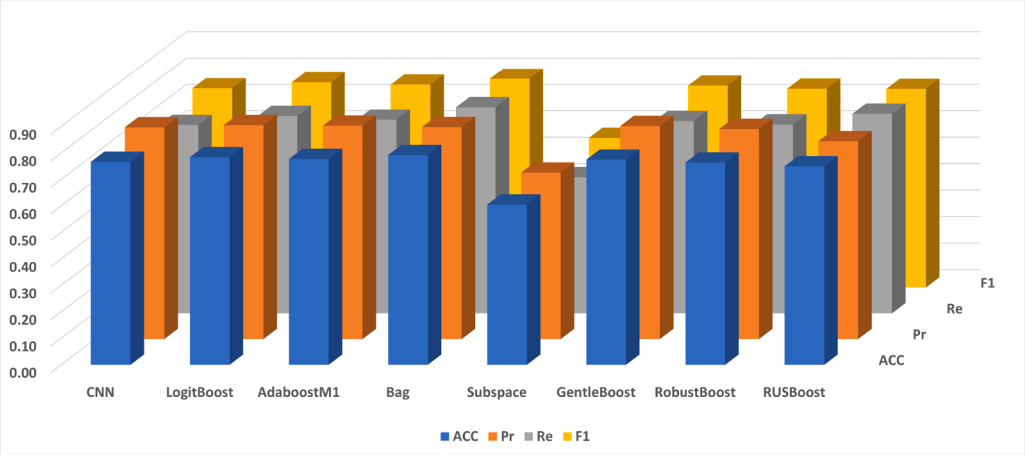


Fig. 11. Efficiency of the binary AED systems using the Log-Mel spectrogram feature (Table 4).

stable than those of the other methods. To validate this claim, a similar analysis to the previous one was conducted using an ANOVA test under the same conditions. Regarding repeatability and reproducibility of results, ANOVA confirmed the significance of differences between Bag and RUSBoost across classifiers and rejected differences within a given classifier. Also, ANOVA rejected the difference between the Bag and the RUSBoost classifiers. As a conclusion, the Bag-type classifier was selected for the current study due to its higher and faster training speed. Table 5 presents the results of the best performing classifiers with standard deviation values obtained from 10 runs.

The low standard deviation values across all metrics confirm the stability and reliability of the proposed approach. To assess the reliability and reproducibility of the AED classifiers, each system underwent 10 independent training runs using the same train, test, and validation sets.

Fig. 12 presents the F1 metric with bars representing standard deviation across these runs for all the studied classifiers. The results demonstrate that the Mel + Bag Ensemble not only achieves the highest performance but also maintains efficiency across runs (standard deviation ≤ 0.01), indicating robust, stable behaviour.

Table 5
Performance metrics of best AED classifiers with standard deviation (mean ± std from 10 runs).

Classifier	ACC	Pr	Re	F1
VGG + SVM RBF	0.62 ± 0.01	0.62 ± 0.02	0.65 ± 0.01	0.63 ± 0.01
VGG + Bag Ensemble	0.62 ± 0.02	0.73 ± 0.02	0.36 ± 0.01	0.49 ± 0.02
YAMNet + Bag Ensemble	0.60 ± 0.01	0.61 ± 0.02	0.54 ± 0.02	0.57 ± 0.01
Mel + Bag Ensemble	0.79 ± 0.01	0.80 ± 0.01	0.78 ± 0.01	0.79 ± 0.01

5.3. Hyperparameter selection and validation strategy

Each classifier's internal hyperparameters were determined via a systematic grid search with 5-fold cross-validation on the training dataset.

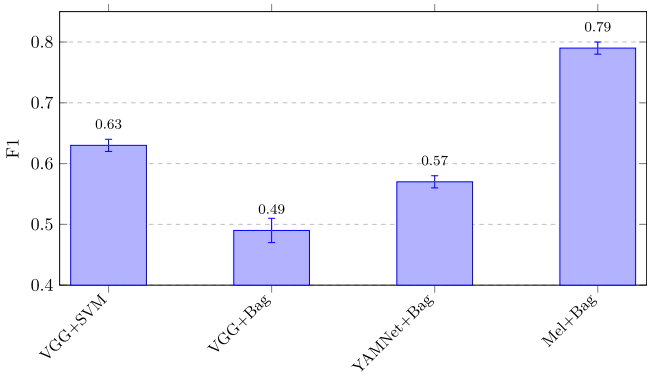


Fig. 12. F1 comparison of the studied AED classifiers with bars showing standard deviation in 10 runs (data from Table 5): Mel + Bag Ensemble achieves the highest F1 of 0.79±0.01 among all tested approaches.

For SVM classifiers, the RBF kernel gamma parameter was explored across the range $[10^{-4}, 10^{-1}]$ with logarithmic spacing, while the regularisation parameter C was searched within $[10^{0}, 10^{3}]$. Polynomial kernel degrees were evaluated from 2 to 5, while linear kernels required only tuning of the C parameter. Optimal parameter combinations were selected based on maximum average F1 across validation folds, with final values reported as: RBF gamma $= 0.01$, $C = 100$ for VGGish features; gamma $= 0.001$, $C = 50$ for YAMNet features. Ensemble learning methods employed MATLAB's Statistics and Machine Learning Toolbox default configurations using the fitcensemble function with the following specifications: Bag ensemble used 30 learners with bootstrap aggregation, AdaBoost M1 applied 100 weak learners with learning rate 0.1, RUSBoost combined random undersampling with boosting using 100 learners and learning rate 0.1, LogitBoost employed 100 logistic regression weak learners, Subspace ensemble used 30 learners with random subspace dimension of 50% of features, GentleBoost applied 100 gentle boosting iterations, and RobustBoost employed 100 robust boosting iterations. These ensemble configurations were adopted without extensive tuning due to their established robustness across diverse classification tasks, prioritising computational efficiency over marginal performance gains from hyperparameter optimisation. A preliminary comparison of default versus grid-searched ensemble parameters on a 20% validation subset showed less than 1% variation in F1 score, justifying the default parameter approach.

CNN architecture hyperparameters were optimised through manual search guided by validation performance. The Adam optimiser learning rate was explored across $[10^{-4}, 10^{-1}]$ in logarithmic steps, with 0.01 yielding optimal convergence. The batch size was evaluated at 32, 64, and 128, with 64 selected based on training stability. The number of convolutional filters was tested at 10, 20, and 30, with 20 filters providing adequate feature extraction without overfitting. Training proceeded for a maximum of 50 epochs with early stopping based on validation loss plateau (patience of 5 epochs), typically converging within 30 epochs. Cross-validation was performed exclusively on training data to prevent information leakage from test and validation sets. The dataset split follows the original partition provided in Castorena et al. (2024a), with 14,952 audio files for training, 1996 for testing, and 1991 for validation. Training samples come from entirely separate audio files compared to test and validation samples, so no acoustic content appears in more than one partition. All classifiers were initialised with MATLAB's default random seed (rng $= 0$) to allow exact reproduction of results. The 10 repeated runs reported in Table 5 used seeds 0 through 9.

All reported metrics, Accuracy, Precision, Recall, and F1, represent deterministic values computed on held-out test and validation sets after model training completion, not averages across cross-validation folds. Confidence intervals were not computed due to the single train-test split design inherent in the provided dataset structure (Castorena et al., 2024a), which pre-allocated specific audio files to the training, testing, and validation partitions. Repeated random sampling was not feasible, as it would introduce data leakage by allowing the same acoustic content to appear in both the training and evaluation sets.

Post-processing hyperparameters (binomial probability threshold of 5 detections in 10 consecutive frames) were derived analytically from the acoustic properties of horn events, rather than being learned from data. Analysis of ground truth labels revealed minimum horn durations exceeding 100 ms, corresponding to at least 10 consecutive 1-second frames with 0.1-second hop length. The majority voting threshold of 5/10 frames strikes a balance between false positive suppression and detection preservation, with the specific value validated through error analysis on the training set.

5.4. Ablation study of AED step

Because the window length of the Log-Mel spectrogram affects both the size of the input features and the computational cost of the AED step, the first step of the ablation study focuses on the effect of window length on the AED step's efficiency. The window length was increased from 0.1 to 1 second in increments of 0.1. Changing the window length also resulted in slight changes in the number of samples. The results are given in Table 6, which shows that while the window length increases from 0.1 second to 1 second, ACC improves slightly, Pr decreases, while Re and F1 show considerable improvement. In fact, using a longer time window helps the system detect horn events better. This reduces FN and brings Pr and Re parameters closer together. At a window length of 0.9 seconds, the results become fairly stable and balanced across all metrics. The 900 ms window performs very similarly to the 1-second window length shown in Table 4, which uses a 1-second window length for extracting Log-Mel Spectrogram.

To confirm this result quantitatively, an ANOVA test was performed under the same conditions as the AED step. In terms of repeatability and reproducibility of results, ANOVA confirmed the significance of differences between window lengths higher than 0.7 seconds (0.8, 0.9, and 1 seconds) and other values, while rejecting the difference between results within a given window length. Additionally, ANOVA rejected the difference between the results of the 0.8, 0.9, and 1-second window lengths. Since the extracted features will also be used in the AEC step, a 1-second window length was chosen as optimal. The 1-second window results presented in Table 4 show performance comparable to that of the 900-ms window in Table 6, while providing better alignment with the requirements of VGGish and YAMNet. It is worth noting that the VGGish and YAMNet architectures have a fixed input length of 0.975 seconds, set during their original training on the AudioSet dataset. Changing this window would require retraining both networks from scratch on the full AudioSet corpus, which falls outside the scope of this study. For the Log-Mel spectrogram path, Table 6 presents a full sensitivity analysis across ten window lengths (from 100 ms to 900 ms in 100 ms steps), covering

Table 6
The effect of the Log-Mel spectrogram window length on AED systems.

			Test								Validation							
			TN	FP	FN	TP	ACC	Pr	Re	F1	TN	FP	FN	TP	ACC	Pr	Re	F1
	CNN	-	149,726	17,274	75,363	91,637	0.72	0.84	0.55	0.66	145,951	16,839	73,463	89,327	0.71	0.81	0.54	0.65
100 ms	Ens.	LogitBoost	147,731	19,269	77,043	89,957	0.71	0.82	0.54	0.65	144,007	18,783	75,101	87,689	0.70	0.80	0.53	0.64
		AdaboostM1	147,388	19,612	77,745	89,255	0.71	0.82	0.53	0.65	143,672	19,118	75,785	87,005	0.71	0.84	0.52	0.64
		Bag	146,797	20,203	74,505	92,495	0.72	0.82	0.55	0.66	143,097	19,693	72,627	90,163	0.71	0.80	0.57	0.66
		Subspace	138,559	28,441	97,064	69,936	0.62	0.71	0.42	0.53	135,066	27,724	94,617	68,173	0.62	0.70	0.43	0.53
		GentleBoost	147,387	19,613	80,672	86,328	0.70	0.81	0.52	0.63	143,671	19,119	78,638	84,152	0.71	0.83	0.52	0.64
		RobustBoost	146,929	20,071	78,263	88,737	0.71	0.82	0.53	0.64	143,225	19,565	76,290	86,500	0.70	0.80	0.53	0.64
		RUSBoost	136,486	30,514	71,457	95,543	0.69	0.76	0.57	0.65	133,045	29,745	69,656	93,134	0.70	0.77	0.58	0.66
	CNN	-	144,933	19,291	71,538	92,686	0.72	0.83	0.56	0.67	141,364	18,816	69,776	90,404	0.73	0.84	0.56	0.67
200 ms	Ens.	LogitBoost	143,229	20,995	71,366	92,858	0.72	0.82	0.57	0.67	139,702	20,478	69,609	90,571	0.71	0.81	0.55	0.66
		AdaboostM1	142,909	21,315	72,808	91,416	0.71	0.81	0.56	0.66	139,390	20,790	71,015	89,165	0.72	0.83	0.56	0.67
		Bag	142,926	21,298	69,070	95,154	0.72	0.82	0.58	0.68	139,406	20,774	67,369	92,811	0.72	0.80	0.59	0.68
		Subspace	133,848	30,376	93,572	70,652	0.62	0.70	0.43	0.53	130,552	29,628	91,268	68,912	0.62	0.70	0.43	0.53
		GentleBoost	143,748	20,476	74,776	89,448	0.71	0.81	0.54	0.65	140,208	19,972	72,935	87,245	0.71	0.81	0.54	0.65
		RobustBoost	142,104	22,120	72,721	91,503	0.71	0.81	0.56	0.66	138,605	21,575	70,930	89,250	0.71	0.81	0.55	0.65
		RUSBoost	132,334	31,890	67,200	97,024	0.70	0.75	0.59	0.66	129,075	31,105	65,545	94,635	0.69	0.74	0.59	0.66
	CNN	-	142,849	18,630	67,147	94,332	0.73	0.84	0.58	0.69	139,410	18,182	65,531	92,061	0.73	0.83	0.59	0.69
300 ms	Ens.	LogitBoost	140,937	20,542	66,998	94,481	0.73	0.82	0.59	0.68	137,544	20,048	65,385	92,207	0.73	0.80	0.59	0.68
		AdaboostM1	141,204	20,275	67,365	94,114	0.73	0.82	0.58	0.68	137,805	19,787	65,743	91,849	0.74	0.85	0.59	0.69
		Bag	137,642	23,837	62,941	98,538	0.73	0.81	0.61	0.69	134,329	23,263	61,426	96,166	0.73	0.82	0.60	0.69
		Subspace	127,416	34,063	89,975	71,504	0.62	0.68	0.44	0.54	124,349	33,243	87,809	69,783	0.61	0.67	0.44	0.53
		GentleBoost	141,356	20,123	70,486	90,993	0.72	0.82	0.56	0.67	137,953	19,639	68,789	88,803	0.72	0.81	0.56	0.67
		RobustBoost	140,111	21,368	67,757	93,722	0.72	0.81	0.58	0.68	136,738	20,854	66,126	91,466	0.72	0.81	0.58	0.67
		RUSBoost	129,576	31,903	61,835	99,644	0.71	0.76	0.62	0.68	126,457	31,135	60,347	97,245	0.71	0.75	0.62	0.68
	CNN	-	139,056	19,706	62,545	96,217	0.74	0.83	0.61	0.70	135,789	19,243	61,076	93,956	0.75	0.85	0.61	0.71
400 ms	Ens.	LogitBoost	136,853	21,909	60,364	98,398	0.74	0.82	0.62	0.71	133,638	21,394	58,946	96,086	0.74	0.83	0.61	0.70
		AdaboostM1	137,005	21,757	63,025	95,737	0.73	0.81	0.60	0.69	133,786	21,246	61,544	93,488	0.73	0.83	0.59	0.69
		Bag	134,641	24,121	57,916	100,846	0.74	0.81	0.64	0.71	131,478	23,554	56,555	98,477	0.75	0.82	0.65	0.72
		Subspace	124,183	34,579	86,840	71,922	0.62	0.68	0.45	0.54	121,265	33,767	84,800	70,232	0.63	0.69	0.46	0.55
		GentleBoost	135,291	23,471	63,676	95,086	0.73	0.80	0.60	0.69	132,112	22,920	62,180	92,852	0.72	0.79	0.60	0.68
		RobustBoost	134,852	23,910	62,748	96,014	0.73	0.80	0.60	0.69	131,684	23,348	61,274	93,758	0.71	0.78	0.59	0.68
		RUSBoost	126,366	32,396	56,159	102,603	0.72	0.76	0.65	0.70	123,397	31,635	54,840	100,192	0.71	0.74	0.63	0.68
	CNN	-	135,409	20,664	57,027	99,046	0.75	0.83	0.63	0.72	132,306	20,191	55,720	96,777	0.75	0.85	0.62	0.72
500 ms	Ens.	LogitBoost	132,666	23,407	55,249	100,824	0.75	0.81	0.65	0.72	129,626	22,871	53,983	98,514	0.74	0.80	0.65	0.72
		AdaboostM1	134,072	22,001	57,505	98,568	0.75	0.82	0.63	0.71	131,000	21,497	56,187	96,310	0.75	0.84	0.62	0.71
		Bag	132,411	23,662	52,685	103,388	0.76	0.81	0.66	0.73	129,377	23,120	51,478	101,019	0.76	0.81	0.67	0.74
		Subspace	119,355	36,718	83,081	72,992	0.62	0.67	0.47	0.55	116,620	35,877	81,177	71,320	0.61	0.66	0.46	0.54
		GentleBoost	133,811	22,262	59,327	96,746	0.74	0.81	0.62	0.70	130,745	21,752	57,968	94,529	0.74	0.81	0.61	0.70
		RobustBoost	131,445	24,628	59,879	96,194	0.73	0.80	0.62	0.69	128,433	24,064	58,507	93,990	0.73	0.80	0.60	0.69
		RUSBoost	121,662	34,411	52,214	103,859	0.72	0.75	0.67	0.71	118,874	33,623	51,018	101,479	0.73	0.76	0.66	0.71
	CNN	-	129,726	23,683	53,847	99,562	0.75	0.81	0.65	0.72	126,831	23,155	52,646	97,340	0.75	0.81	0.66	0.73
600 ms	Ens.	LogitBoost	130,295	23,114	51,678	101,731	0.76	0.81	0.66	0.73	127,388	22,598	50,525	99,461	0.76	0.82	0.66	0.73
		AdaboostM1	129,155	24,254	53,520	99,889	0.75	0.80	0.65	0.72	126,273	23,713	52,326	97,660	0.74	0.79	0.66	0.71
		Bag	126,803	26,606	46,247	107,162	0.76	0.80	0.70	0.75	123,974	26,012	45,215	104,771	0.77	0.82	0.70	0.76
		Subspace	114,502	38,907	79,587	73,822	0.61	0.65	0.48	0.55	111,947	38,039	77,811	72,175	0.60	0.64	0.47	0.54
		GentleBoost	130,381	23,028	54,876	98,533	0.75	0.81	0.64	0.72	127,472	22,514	53,652	96,334	0.75	0.83	0.63	0.72
		RobustBoost	129,526	23,883	53,861	99,548	0.75	0.81	0.65	0.72	126,636	23,350	52,659	97,327	0.74	0.81	0.64	0.71
		RUSBoost	118,793	34,616	47,370	106,039	0.73	0.75	0.69	0.72	116,142	33,844	46,313	103,673	0.73	0.76	0.69	0.72
	CNN	–	127,225	23,542	50,118	100,649	0.76	0.81	0.67	0.73	124,462	23,031	49,030	98,463	0.75	0.79	0.66	0.72
700 ms	Ens.	LogitBoost	126,317	24,450	45,551	105,216	0.77	0.81	0.70	0.75	123,574	23,919	44,562	102,931	0.77	0.80	0.71	0.75
		AdaboostM1	126,044	24,723	48,744	102,023	0.76	0.80	0.68	0.74	123,307	24,186	47,685	99,808	0.75	0.79	0.66	0.72
		Bag	125,531	25,236	41,277	109,490	0.78	0.81	0.73	0.77	122,805	24,688	40,381	107,112	0.78	0.81	0.74	0.77
		Subspace	110,326	40,441	76,508	74,259	0.61	0.65	0.49	0.56	107,930	39,563	74,847	72,646	0.61	0.65	0.50	0.56
		GentleBoost	125,501	25,266	49,198	101,569	0.75	0.80	0.67	0.73	122,776	24,717	48,130	99,363	0.76	0.80	0.69	0.74
		RobustBoost	124,683	26,084	50,975	99,792	0.74	0.79	0.66	0.72	121,975	25,518	49,868	97,625	0.74	0.80	0.66	0.72
		RUSBoost	116,050	34,717	43,331	107,436	0.74	0.76	0.71	0.73	113,530	33,963	42,390	105,103	0.73	0.74	0.70	0.72
	CNN	–	121,993	26,163	45,911	102,245	0.76	0.80	0.69	0.74	119,420	25,611	44,943	100,088	0.76	0.80	0.70	0.75
800 ms	Ens.	LogitBoost	122,627	25,529	40,973	107,183	0.78	0.81	0.72	0.76	120,040	24,991	40,109	104,922	0.77	0.80	0.71	0.75
		AdaboostM1	123,742	24,414	44,354	103,802	0.77	0.81	0.70	0.75	121,132	23,899	43,418	101,613	0.76	0.79	0.70	0.74
		Bag	120,514	27,642	36,256	111,900	0.78	0.80	0.76	0.78	117,972	27,059	35,491	109,540	0.79	0.82	0.75	0.78
		Subspace	106,544	41,612	74,433	73,723	0.61	0.64	0.50	0.56	104,297	40,734	72,863	72,168	0.60	0.63	0.49	0.55
		GentleBoost	123,102	25,054	43,748	104,408	0.77	0.81	0.70	0.75	120,505	24,526	42,825	102,206	0.77	0.81	0.72	0.76
		RobustBoost	121,677	26,479	46,225	101,931	0.75	0.79	0.69	0.74	119,111	25,920	45,250	99,781	0.76	0.81	0.68	0.74
		RUSBoost	111,975	36,181	39,406	108,750	0.74	0.75	0.73	0.74	109,613	35,418	38,575	106,456	0.76	0.77	0.75	0.76
	CNN	–	120,760	24,812	41,107	104,465	0.77	0.81	0.72	0.76	118,287	24,304	40,265	102,326	0.76	0.79	0.71	0.75
900 ms	Ens.	LogitBoost	120,152	25,420	38,148	107,424	0.78	0.81	0.74	0.77	117,692	24,899	37,367	105,224	0.78	0.81	0.73	0.77
		AdaboostM1	120,039	25,533	38,460	107,112	0.78	0.81	0.74	0.77	117,581	25,010	37,672	104,919	0.77	0.79	0.73	0.76
		Bag	117,338	28,234	32,341	113,231	0.79	0.80	0.78	0.79	114,935	27,656	31,679	110,912	0.77	0.78	0.76	0.77
		Subspace	101,317	44,255	71,381	74,191	0.60	0.63	0.51	0.56	99,242	43,349	69,919	72,672	0.60	0.63	0.50	0.56
		GentleBoost	119,397	26,175	40,706	104,866	0.77	0.80	0.72	0.76	116,952	25,639	39,872	102,719	0.77	0.80	0.71	0.75
		RobustBoost	119,587	25,985	40,950	104,622	0.77	0.80	0.72	0.76	117,138	25,453	40,111	102,480	0.76	0.79	0.72	0.75
		RUSBoost	108,495	37,077	35,889	109,683	0.75	0.75	0.75	0.75	106,273	36,318	35,154	107,437	0.75	0.75	0.77	0.76

Table 7
AED performance before and after post-processing (Mel + Bag Ensemble).

Set	Stage	ACC	Pr	Re	F1
Test	Before post-processing	0.79	0.80	0.78	0.79
	After post-processing	0.97	0.97	0.96	0.97
Validation	Before post-processing	0.78	0.79	0.76	0.77
	After post-processing	0.96	0.97	0.95	0.96

windows both shorter and longer than the 0.5 seconds suggested as an alternative. The results confirm that performance stabilises at window lengths above 700 ms, with 0.8, 0.9, and 1-second windows showing no statistically significant differences (ANOVA, $p > .05$).

5.5. Post-processing

Analysis of the Horns database revealed that horn signals last more than 100 ms. Given the 1-second window length and 0.1-second hop length, a horn should be detected across 10 consecutive frames. This characteristic enables error reduction through a post-processing step based on the binomial probability distribution: a horn label is assigned only if detection occurs in at least five of 10 frames. The error probability corresponds to six or more false detections in a 10-frame sequence. Let k denote wrong detections, p the ACC of the AED system, and $q = 1 - p$ the error probability for 1 frame. With the best ACC = 0.79 (Table 4), the theoretical post-processing error under the assumption of independent frame errors is:

$$
\begin{aligned}
&p = .79, \quad q = 0.21,\\
&\binom{10}{k} = \frac{10!}{k!(10-k)!},\\
&P(X = k) = \binom{10}{k} q^k p^{10-k},\\
&P(X > 5) = \sum_{k=6}^{10} P(X = k) \rightarrow \sum_{k=6}^{10} \binom{10}{k} (0.21)^k (0.79)^{10-k} \approx 0.0082.
\end{aligned}
\tag{10}
$$

Eq. (10) yields a theoretical upper bound of approximately 99.2% for AED accuracy after post-processing. However, in practice, consecutive frames share a large portion of their content due to the 0.1-second hop length within a 1-second window, which means the independence assumption holds only approximately. To obtain reliable performance figures, the post-processing step was applied to the full test and validation sets, and the actual evaluation metrics were measured directly. Table 7 presents these results.

As shown in Table 7, the measured F1 after post-processing reaches 0.97 on the test set and 0.96 on the validation set. These values fall below the theoretical upper bound of Eq. (10), which is expected given the overlap between adjacent frames. Nevertheless, the post-processing step produces substantial improvements across all metrics, raising F1 by 0.18 on the test set and 0.19 on the validation set. This confirms that the binomial majority voting scheme is an effective and computationally inexpensive way to reduce frame-level errors in the AED step.

5.6. The ablation study of AEC step

Various scenarios were examined regarding the AEC step (Fig. 3). Four binary SVM classifiers with RBF kernels were employed to separate the four horn classes, and the input feature vector was considered to include a maximum of 128 features, corresponding to the full VGGish embedding dimension. The RBF kernel was chosen for its superior performance compared to other SVM kernels. The input acoustic signal is taken from the AED step's outputs, which are labelled 'horn'. The results for the four classes, including Boat, Car, Truck, and Train Horn, are presented in Figs. 13 to 16. In each figure, four evaluation metrics, ACC, Pr, Re, and F1, are plotted simultaneously to clearly show the trade-off between performance vs feature count in each scenario. Meanwhile, comparing the plots reveals the effectiveness of different features in the AEC step.

Table 8
AEC performance before and after post-processing per horn class (VGGish + mRMR + SVM, test set).

Horn class	Stage	ACC	Pr	Re	F1
Boat	Before post-processing	0.89	0.85	0.80	0.82
	After post-processing	0.97	0.96	0.95	0.96
Car	Before post-processing	0.90	0.86	0.81	0.83
	After post-processing	0.97	0.97	0.96	0.97
Train	Before post-processing	0.98	0.90	0.86	0.88
	After post-processing	0.99	0.98	0.97	0.98
Truck	Before post-processing	0.96	0.88	0.83	0.85
	After post-processing	0.98	0.97	0.97	0.97

Fig. 13 demonstrates that when VGGish has been employed as a feature extractor, the AED showed better performance in the boat class, and YAMNet and Log-Mel spectrogram features, separately or in combination with VGGish, did not improve the AED's performance. In VGGish, increasing the feature count has improved performance, though the improvement is minimal for feature counts greater than 100. With more than 100 features, the performance of all classifiers becomes relatively stable and acceptable. However, the significant differences between Re, Pr, and F1, and between ACC across all plots, indicate the effect of unbalanced training for this class, which is unacceptable.

In Figs. 15 and 16, where the data are less imbalanced, the systems' performance across all four metrics is noticeably better than that of the Boat and Car classes. In these two cases, ACC exceeds 98%, and the F1 parameter exceeds 0.8; additionally, the differences between Re, Pr, and ACC are decreasing, indicating proper system training progress. Like boat and car classes, the VGGish is the best feature. In the next step, to improve performance for the Car and Boat classes (Figs. 13 and 14), the VGGish approach in these two classes was retrained using augmented data. The results are demonstrated in Fig. 17. The significant difference between Re, F1, and ACC in all cases indicates the effect of unbalanced training in the boat class (Table 2), which is undesirable.

Fig. 14 exhibits a pattern similar to that of Fig. 13 for the performance of different features, with VGGish features performing better than the others. After data augmentation, the VGGish feature-based AEC system showed improved accuracy for the Boat and Car classes.

As shown in Fig. 17, the evaluation metrics converged, with all four reaching values above 0.8. To confirm that this gain comes from the balanced training data rather than from a change in the model itself, the same VGGish + mRMR + SVM pipeline was used in both cases with identical hyperparameters.

The only difference was the composition of the training set. Before augmentation (Fig. 13A for Boat, Fig. 14A for Car), the metrics showed a clear gap between ACC and the other three indicators, which is a typical sign of class imbalance affecting the classifier. After augmentation (Fig. 17), all four metrics rose and converged, meaning the classifier was no longer biased towards the majority class. This comparison confirms that the augmentation step was needed specifically for the two most imbalanced classes (Boat and Car), while the Train and Truck classes (Figs. 15A and 16A) already performed well without extra samples because their original class ratios were closer to balanced (Table 2). To further reduce errors across all four classifiers, the same binomial post-processing step from the AED stage (Section 5.5) was applied to the AEC outputs.

Table 8 presents the AEC performance before and after post-processing for each horn class on the test set, using 80 VGGish features selected by mRMR. The results in Table 8 confirm that the binomial post-processing raises the minimum F1 from 0.82 (Boat class) to 0.96 after post-processing. The Train and Truck classes, which had lower class im-

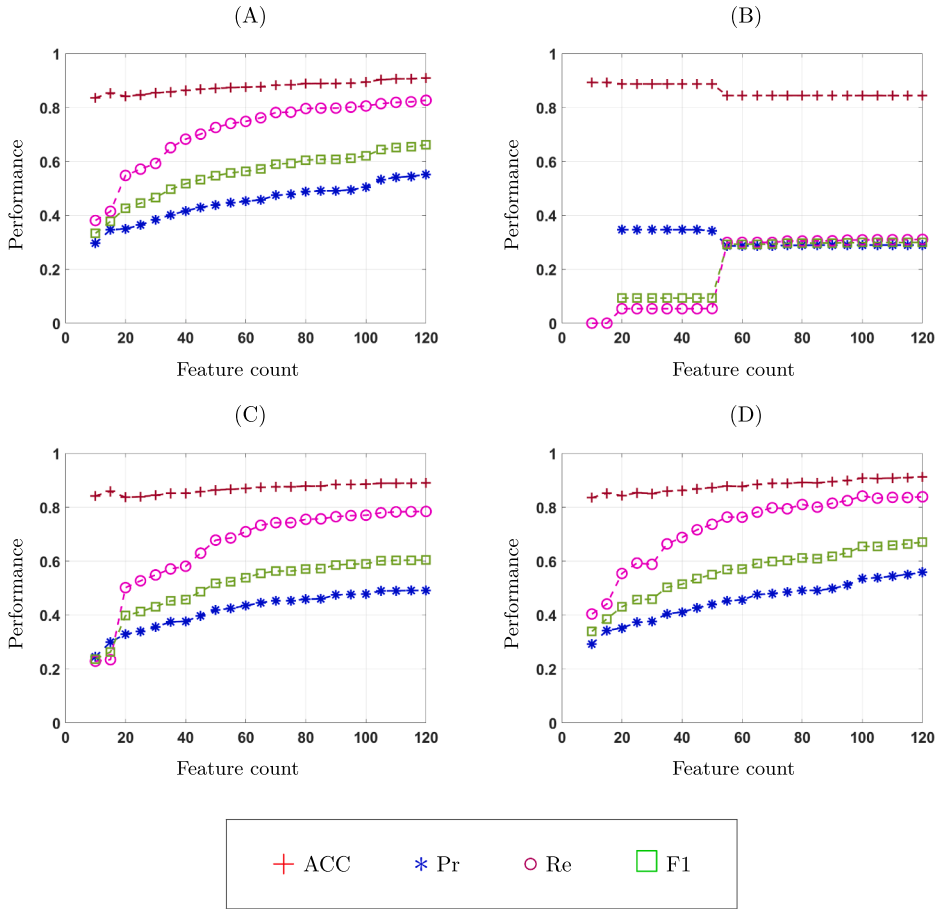


Fig. 13. Efficiency of proposed AEC systems in the Boat class: A - VGGish, B - YAMNet, C - Log-Mel spectrogram, and D - VGGish + YAMNet + Mel.

Table 9
Confusion matrix for the AED step after BPDF post processing (Mel + Bag, test set, row normalised %).

	Predicted Horn	Predicted Non Horn
Actual Horn	**97**	3
Actual Non Horn	2	**98**

Table 10
Confusion matrix for the AEC step after BPDF post processing (VGGish + mRMR + SVM, test set, row normalised %).

	Boat	Car	Train	Truck
Boat	**96**	2	1	1
Car	2	**97**	0	1
Train	0	1	**98**	1
Truck	1	1	1	**97**

balance in the original data, achieve F1 scores of 0.98 and 0.97, respectively. These measured values fall below the theoretical upper bound from Eq. (10) for the same reasons discussed in Section 5.5, but still represent a considerable gain across all four horn categories.

5.7. Confusion matrix analysis

To give a clearer picture of how the proposed framework distributes its predictions across classes, Tables 9 and 10 present the confusion matrices for the AED and AEC steps after BPDF post-processing, expressed as row-normalised percentages. Each row represents the true class, and each column represents the predicted class, so the diagonal entries correspond to the per-class recall. For the AED step, Table 9 shows that the Mel + Bag classifier with BPDF post processing correctly identifies 97% of horn event frames and 98% of non-horn frames in the test set. Only 3% of actual horn frames go undetected, and only 2% of non-horn frames trigger a false alarm. These figures are consistent with the F1 score of 0.97 reported in Table 7 and confirm that the post-processing stage removes the majority of isolated misclassifications that occur at the single frame level. For the AEC step, Table 10 presents the four-class confusion matrix for the VGGish + mRMR + SVM classifier after BPDF post-processing.

All four horn classes achieve diagonal values of 96% or above, which is consistent with the per-class F1 scores in Table 8. The small number of misclassifications that do occur follow an expected pattern: the Boat and Car classes show slightly more confusion with each other than with Train or Truck, which matches the spectral overlap between these two horn types observed in Section 5.1. The Train class achieves the highest recall at 98%, reflecting its distinct acoustic signature, with a longer duration and lower fundamental frequency than the other horn types. The strong diagonal dominance in both matrices confirms that the two-stage framework produces reliable predictions with very few misclassifications, both at the detection level and at the classification level.
It is also worth considering how the system behaves under different noise conditions. The Castorena database (Castorena et al., 2024a) used in this study consists of recordings captured in real urban settings with naturally varying levels of background noise from traffic, wind, crowd activity, and other environmental sources. The augmentation pipeline described in Section 4.3 further increases this variation, since SCAPER generates new training samples by mixing horn events with different

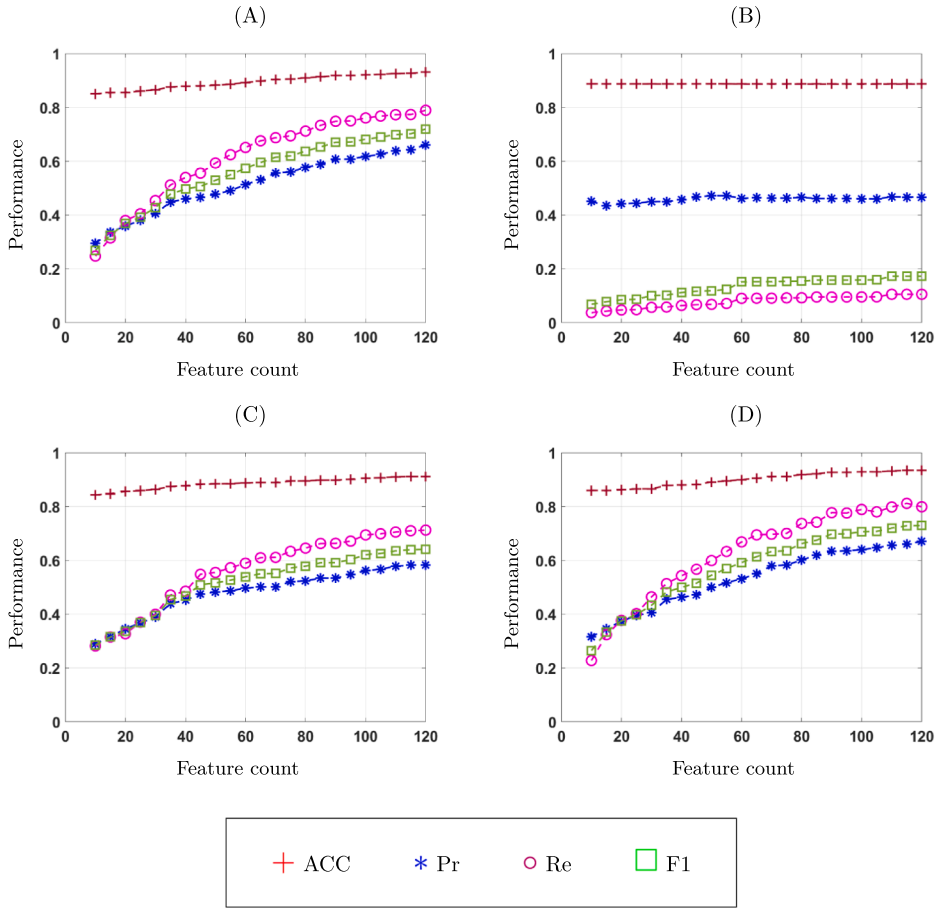


Fig. 14. Efficiency of proposed AEC systems in the Car class: A - VGGish, B - YAMNet, C - Log-Mel spectrogram, and D - VGGish + YAMNet + Mel.

background soundscapes at a range of signal-to-noise ratios. As a result, both the training and test sets reflect a broad spread of noise conditions rather than a single controlled environment. The F1 scores of 0.97 for AED (Table 7) and 0.96 or higher for all four AEC classes (Table 10) were observed across this full range, suggesting that the proposed framework handles realistic noise variation without a substantial drop in performance. A formal sensitivity study at fixed SNR levels (e.g., 0, 6, and 12 dB) could provide a finer breakdown of where the system begins to struggle, and this remains a useful direction for future research. However, for the intended deployment in automotive ADAS, the performance across naturally varying urban noise conditions is the more relevant test, and the results presented here show that the method meets this requirement. Another question that deserves attention is how sensitive AEC performance is to the number of features selected by mRMR.

In this study, the mRMR algorithm ranked all 128 VGGish embedding dimensions by their relevance to the target classes, and the top 80 were kept for classification. The choice of 80 features can be evaluated against the other feature configurations already tested in this work. The VGGish scheme without mRMR uses the full 128-dimensional embedding, while the Combined scheme concatenates VGGish, YAMNet, and Log-Mel features into a vector that is several times larger. As Fig. 13 through Fig. 16 show, the VGGish + mRMR scheme with 80 features matches or exceeds the full 128-feature VGGish scheme across all four horn classes, and it also outperforms the much larger Combined feature set. This means that keeping roughly 60% of the original VGGish dimensions is enough to capture the information that matters for distinguishing horn types, and that adding more features beyond this point does not help. The mRMR ranking itself provides further evidence: the mutual information scores of the top 80 features drop off sharply after a certain point, which indicates that the remaining 48 dimensions carry mostly redundant or irrelevant information. A finer sweep across different values of the selected feature count (e.g., at steps of 10 from 20 to 128) could map the exact point at which performance begins to plateau, which is noted as a possible extension. Nevertheless, the existing comparison between 80 mRMR-selected features and the full 128 features already confirms that the chosen operating point lies in a region where performance is strong and stable. A related question is how the 80-feature subset would perform on acoustic data from environments not included in the original training. The mRMR ranking depends on the statistical properties of the Castorena training set (Castorena et al., 2024a), so the 80 indices it selects were tuned to that database. If the recording conditions change substantially, for example, data collected in a different city or with different vehicle categories, the distribution of VGGish embeddings may shift, and a new mRMR ranking pass on a small sample of labelled data from the target environment would be the right approach. This step completes in under 10 seconds on a standard desktop machine, requires no modification to the VGGish network or the SVM classifiers, and produces a new set of 80 indices to replace the old ones.

5.8. Discussion

The weaker AED results from VGGish (F1 = 0.63 with SVM RBF, Table 4) and YAMNet (F1 = 0.63 with RUSBoost, Table 4) may appear to contradict the claim that these models handle noisy conditions well. However, the AED task is a binary classification problem in which the classifier must separate one target event from 45 other acoustic categories, each mixed with background noise. The 128-dimensional VGGish embedding was originally trained to distinguish among hundreds of sound classes in AudioSet, so it spreads its representational capacity across many event types rather than focusing on the horn vs. no-horn

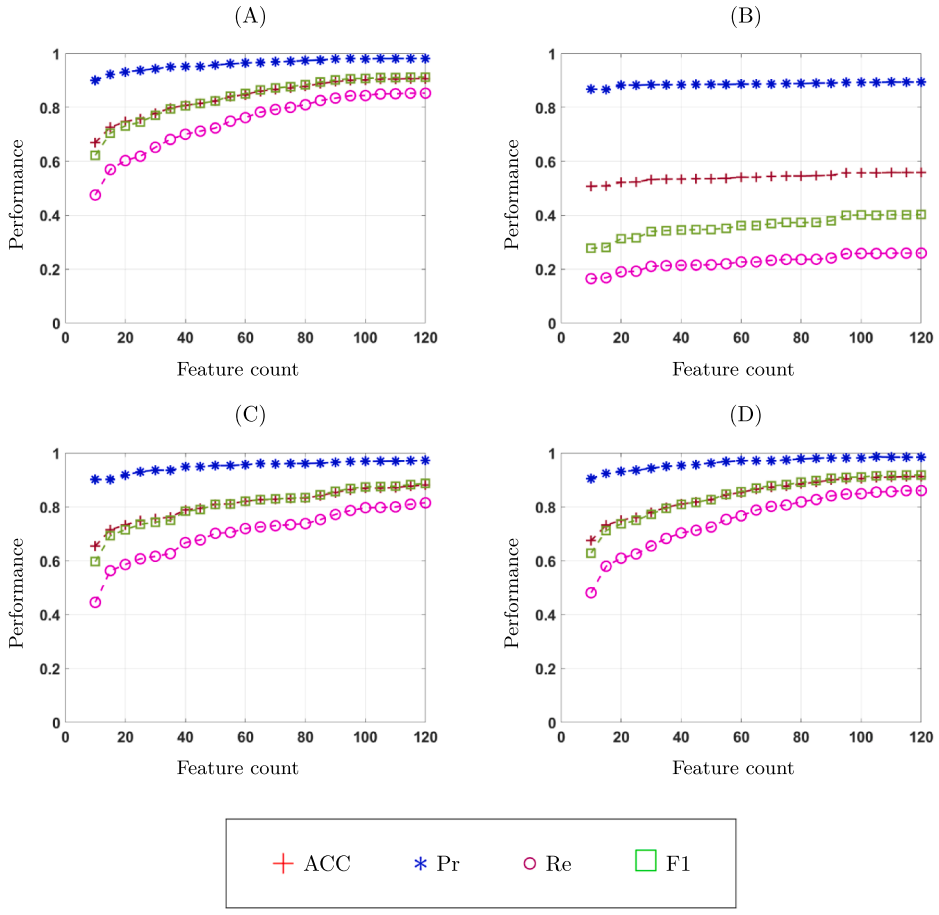


Fig. 15. Efficiency of proposed AEC systems in the Train class: A - VGGish, B - YAMNet, C - Log-Mel spectrogram, and D - VGGish + YAMNet + Mel.

boundary. The Log-Mel spectrogram, on the other hand, preserves the full time-frequency detail of the input frame, which gives the Bag ensemble enough local information to draw a cleaner decision boundary for this specific binary task. In the AEC step, the situation reverses: once the input is already known to be a horn, the VGGish embedding becomes the strongest feature because its learned representations capture the finer differences between horn types (Figs. 13–16).

This pattern confirms that the strength of VGGish lies in classification among similar events rather than in broad detection against a diverse background. A complete analysis of how the Log-Mel spectrogram's window length affects AED performance (Table 6) was done as an ablation study. The results showed that 0.975 seconds is the best choice when considering both suitable performance and its application in the AEC step. Among the available classifiers, the Ensemble Bag classifier delivered the best results across all Precision, Accuracy, Recall, and F1 metrics (Fig. 11). Feature selection was not performed during the AED step because the chosen Log-Mel spectrogram feature is the foundation for extracting VGGish and YAMNet features.

In the AEC step for horn type classification, the proposed approach, shown in Fig. 3, used an SVM classifier with a one-dimensional input structure. This approach converts the Log-Mel spectrogram to a one-dimensional vector after passing through the mRMR feature selection block. After examination, it was found that the VGGish feature performs much better than other approaches across all four horn classes (Figs. 13–16). Changing the window length at this step was impossible, as the pretrained VGGish network was used. Therefore, the ablation study focused on the number of input features to the classifier.

The results showed that VGGish features outperformed all other features across every horn class, and that AEC performance changed very little when using 80 or more features. This means that approximately 40% of the input feature vector can be removed without noticeably reducing AEC performance, compared to the case where the mRMR step is discarded, thereby helping lower the system's power consumption and computational complexity. To validate the above inferences in the AEC step, a similar analysis to the AED step was conducted using an ANOVA test under the same conditions. In terms of repeatability and reproducibility, ANOVA confirmed the significance of differences between pre-trained VGGish and other features. Also, ANOVA rejected the null hypothesis of no difference within a given classifier.

5.9. Comparison with previous methods

Based on our knowledge, three studies have been conducted using the database (Castorena et al., 2024a). A maximum F1 of 0.65 was reached using YOLOv5s with Mel spectrogram input (Castorena et al., 2024a). The performance of several 2D deep learning classification methods on this database was evaluated, and by making some modifications and adjusting network parameters, an F1 score of 0.68 was achieved (Castorena et al., 2025). Further improvements were achieved by using a newer version of YOLOv5s and incorporating additional processing techniques, such as dynamic quantisation, resulting in a maximum F1 score of 0.81 (Castorena et al., 2024b). Given its shorter feature length and simpler structure, the proposed method is computationally lighter than existing 2D approaches. The post-processing step introduced in Section 5.5 yields substantial gains in both AED and AEC performance without imposing a notable computational burden. As reported in Table 7, the AED stage reaches F1 = 0.97 on the test set after post-processing, up from 0.79 before post-processing. For the AEC stage, Table 8 shows that the minimum F1 across all four horn classes rises from 0.82 to 0.96 after post-processing. These results represent an im-

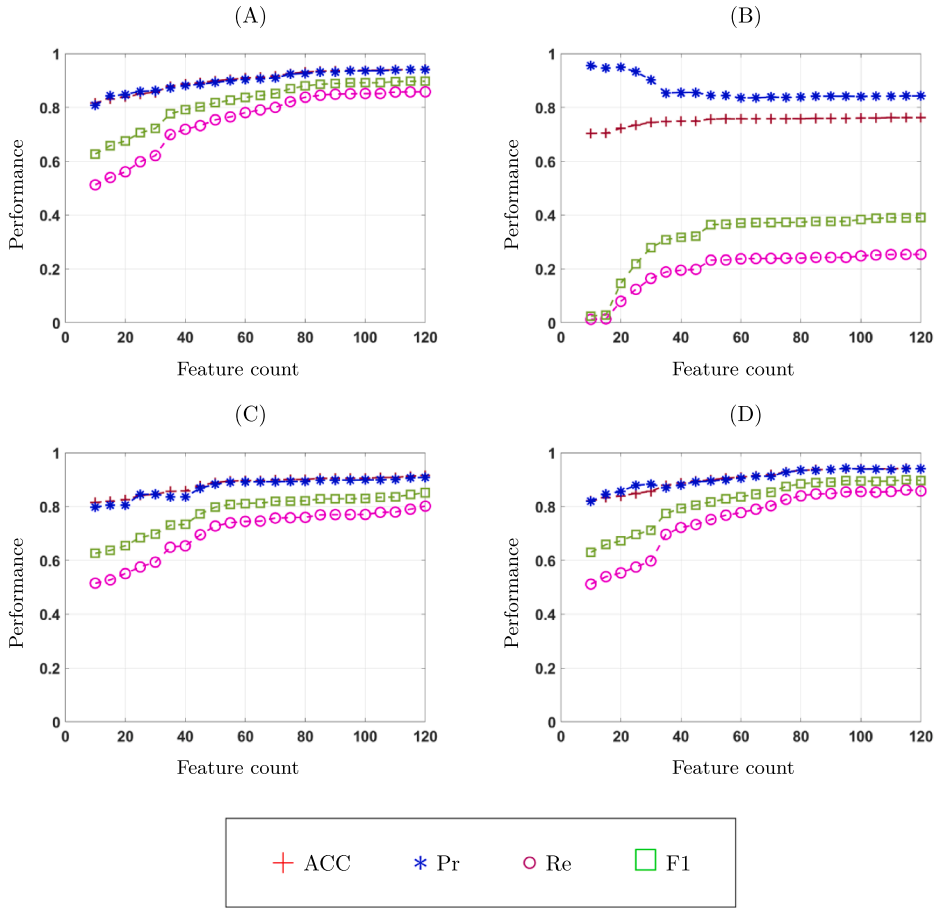


Fig. 16. Efficiency of proposed AEC systems in the Truck class: A - VGGish, B - YAMNet, C - Log-Mel spectrogram, and D - VGGish + YAMNet + Mel.

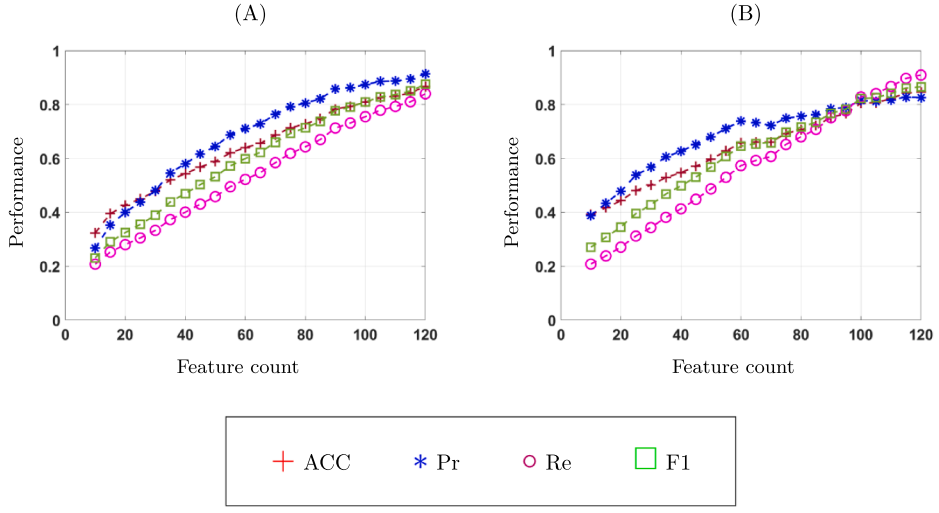


Fig. 17. Efficiency of proposed AEC based on VGGish feature after data augmentation: (A) Boat class and (B) Car class.

provement of more than 15% over the best previously reported F1 score of 0.81 on this database (Castorena et al., 2024b), while using a simpler classification architecture. Table 11 summarises the comparison.

5.10. Computational load and real-time deployment feasibility

Practical implementation in automotive environments requires analysing computational requirements in relation to embedded processor capabilities. The proposed hierarchical architecture introduces computational costs at three stages: feature extraction, classification, and post-processing. The largest computational burden of the proposed method is in the VGGish feature extraction block. The remaining blocks consume less memory and place a lower computational load than VGGish. A log-spectrogram is also extracted using VGGish, which is used in the AED step. VGGish feature extraction, detailed in Section 3.2, represents the primary computational cost. In total, approximately 72 million operations are performed in each VGGish process. Additionally, other computational costs include SVM classifications (dependent on

Table 11
Comparison with previous methods on the same database.

Method	Feature	Classifier	Best F1
Castorena et al. (2024a)	Mel spectrogram	CRNN / YOLOv5s	0.65
Castorena et al. (2025)	Mel spectrogram	2D DL methods	0.68
Castorena et al. (2024b)	Mel spectrogram	YOLOv5s + quantization	0.81
Proposed (before post-proc.)	VGGish + mRMR	SVM (RBF)	0.82
Proposed (after post-proc.)	VGGish + mRMR	SVM (RBF) + BPDF	0.96

the number of support vectors) and binomial post-processing, both of which are negligible. Considering a time step of 0.1 seconds and a 128-dimensional response, and assuming that the post-processing block has sufficient previous samples, in the worst case, the suggested AEDC system can be deployed in real time on a processor with a 1 GHz clock frequency. To put this computational load in perspective, AST contains 87 million trainable parameters across 12 Transformer encoder layers and requires GPU acceleration for both training and inference (Gong et al., 2021).

In contrast, the proposed pipeline needs only the 72 million operations described above and runs on a standard embedded processor without any GPU. For the AEC step, the VGGish feature extractor produces a compact 128-dimensional embedding per 0.975 s window, and after mRMR reduces this to 80 dimensions, the SVM classifier adds a negligible overhead. The total parameter footprint of the inference pipeline remains orders of magnitude below that of AST, since VGGish and YAMNet serve only as frozen feature extractors, and the final classifiers (SVM for AEC, Bagging for AED) carry far fewer tunable weights than a full Transformer encoder stack. This design choice reflects the core goal of this work: to build a system that can operate within the strict power and latency constraints of automotive embedded platforms, rather than to achieve the highest possible accuracy on large-scale general-purpose benchmarks. One practical consideration for deployment is the buffering latency introduced by the 0.975 s analysis window required by VGGish and YAMNet. Since both networks expect a fixed-duration input of approximately one second, the system must collect this amount of audio before it can produce a classification output. When combined with the inference time of roughly 72 ms on a 1 GHz processor, the total response delay from the onset of a horn event to the system output is approximately 1.05 s. It is worth noting that horn detection in an automotive setting serves as a driver awareness function rather than an emergency collision avoidance action. Collision avoidance systems such as autonomous emergency braking operate on much tighter time budgets, typically under 200 ms.

However, advisory alerts that inform the driver about the surrounding acoustic environment can tolerate longer response windows. A delay of roughly 1 second is consistent with the reaction time range that drivers themselves need to perceive and respond to external warnings, which various driving behaviour studies place between 0.7 and 2.5 seconds, depending on the complexity of the traffic situation. Furthermore, the AED step can also operate with the Log-Mel features at shorter window lengths, as shown in Table 6 for durations of 100 to 900 ms. This means that a first-stage detection flag can be raised earlier, and the full classification using VGGish and SVM can proceed in the next analysis window to confirm the horn type. Such a two-pass approach would reduce the perceived latency of the initial alert while still providing the detailed classification output shortly after. One further point that affects how the system is understood from a deployment perspective is the role of mRMR in the inference pipeline.

The mRMR algorithm described in Section 3.6 runs only once during the training phase. It takes the full set of 128 VGGish embedding dimensions across the training data, ranks them by their balance of relevance to the target classes and redundancy among themselves, and produces a fixed, ordered list of the top 80 features. This computation took under 10 seconds on a standard desktop machine with an Intel i7 processor and 16 GB of RAM, because the mutual information calculations involved a 128-column matrix over the training samples and nothing more. Once the ranking is complete, the selected 80 feature indices are stored as a simple lookup list. At inference time, when a new audio frame arrives, and VGGish produces its 128-dimensional embedding, the system extracts the 80 pre-selected dimensions by index and passes them directly to the SVM classifier. No mRMR computation takes place during inference at all.

The additional cost of this extraction step is negligible, amounting to a single array indexing operation that adds no measurable latency to the pipeline. In practice, mRMR is a design-time tool that improves classifier performance by removing noise from the feature space and leaving no computational footprint on the deployed system. Regarding storage, the lookup list of 80 integer indices takes up only a few hundred bytes, which is negligible on any automotive embedded processor. In settings where memory is very tight, the VGGish model weights, which occupy roughly 275 MB, are the dominant cost by far. Any platform that can load and run VGGish inference will have enough memory to store the mRMR index list alongside it, so the lookup approach does not introduce any meaningful memory constraint in practice.

5.11. Practical implications and limitations

The proposed AEDC hierarchical framework offers several practical benefits for driving safety, but it also has limitations that affect real-world applications. The main practical advantages include:

- The proposed horn detection system can recognise warning signals from other vehicles in real time, helping to prevent accidents in smart cars;
- Because it utilises Log-Mel spectrograms and VGGish, the system performs well even in noisy city environments, providing drivers with better awareness of their surroundings;
- Adding the proposed AEDC module to existing ADAS enhances safety, especially when visibility is poor due to fog, at night, or when the view is blocked;
- It offers a more cost-effective alternative to camera-based ADAS systems.

However, there are several limitations:

- The proposed system may not work as well in places with many overlapping sounds, like construction areas or heavy traffic, where horn signals get mixed up with other noises or become distorted;
- Some regions utilise horns for other applications, so the model may need to be adjusted for specific areas to function properly everywhere;
- Running the system in real time requires optimising the model and the hardware;
- The system requires a relatively clean acoustic signal to function properly, so the placement of the microphone and the method used to filter out background noise are crucial;
- Sounds that are not horns but have similar frequency patterns could trigger false alarms, so the feature extraction and classification need fine-tuning.

6. Conclusion

This study presents a cost-effective and efficient AEDC approach for horn signals in driving scenarios. The proposed method includes two AED and AEC steps in a hierarchical structure. In the AED step, the Log-Mel spectrogram feature with a Bag-type Ensemble learning showed the best result. In the AEC step, VGGish features, which are efficient and well established for acoustic signals, are used to classify the horn type. The AEC step classifiers were a set of SVMs with the RBF kernel. Finally, in both AEC and AED steps, a post-processing block based on BPDF was used to correct possible errors in the output. The parameters of the proposed system, both in feature extraction and classification, have been

optimised through a series of simulations to maximise performance. In the AEC step, the mRMR feature selection scheme was employed to reduce the input feature dimension and eliminate redundant features. VGGish and Log-Mel spectrogram methods, which can be implemented at very high speeds and do not require task-specific training on the target dataset, are among the proposed method's strengths. Additionally, having four parallel classifiers enables the AEDC system to detect horn signals simultaneously from multiple sources, which is particularly important in practical scenarios. Simulation results show that the proposed approach achieves F1 = 0.97 in the AED step and a minimum F1 = 0.96 across all horn classes in the AEC step after post-processing, representing an improvement of more than 15% over the latest approaches on the same database while reducing classifier complexity.

Given the importance of driver and passenger safety and security in vehicles, as well as the need to pay attention to environmental sounds in cities and on roads, having an intelligent system like the proposed method, which can detect horn signals and reduce driver distraction, is crucial. Additionally, the proposed AEDC system can be used in smart vehicles to help prevent accidents. Future research directions encompass three primary themes: robustness enhancement, multi-modal integration, and extended acoustic scene understanding. First, sensitivity analysis under diverse environmental noise conditions requires field testing. The impact of acoustically similar signals, such as emergency vehicle sirens, on classification specificity warrants systematic evaluation. Second, integration of the proposed AEDC system with ADAS could improve detection reliability and situational awareness. Third, acoustic scene understanding and classification, which consider the auditory environment, enhance the efficiency of the horn detection system.

CRediT authorship contribution statement

Vahid Hajihashemi: Investigation, Data curation, Software, Formal analysis, Writing – original draft; **Abdorreza Alavi Gharahbagh:** Investigation, Data curation, Software, Formal analysis, Writing – original draft; **José J.M. Machado:** Writing – review & editing; **João Manuel R.S. Tavares:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This article is partially a result of the project "Sensitive Industry", with reference SIFN-01-9999-FN-182852, co-funded by the European Regional Development Fund (ERDF), through the Operational Programme for Competitiveness and Internationalisation (COMPETE 2020), under the PORTUGAL 2020 Partnership Agreement. The first author would like to thank "Fundação para a Ciência e a Tecnologia (FCT)", in Portugal, for his PhD grant, reference 2021.08660.BD.

Data availability

The data used in publicly available.

References

Albadawi, Y., Takruri, M., & Awad, M. (2022). A review of recent developments in driver drowsiness detection systems. *Sensors*, *22*(5), 2069. https://doi.org/10.3390/s22052069.

Alemany, R., Ayuso, M., & Guillén, M. (2013). Impact of road traffic injuries on disability rates and long-term care costs in spain. *Accident Analysis & Prevention*, *60*, 95–102. https://doi.org/10.1016/j.aap.2013.08.016.

Angulo, F., Essid, S., Peeters, G., & Mietlicki, C. (2023). Cosmopolite sound monitoring (cosmo): A study of urban sound event detection systems generalizing to multiple cities. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10095833.

Avila, A., & Pinzón, C. (2024). Comparative analysis of VGGish and YAMNet models for welding defect detection. In *International scientific-technical conference manufacturing* (pp. 184–199). Springer. https://doi.org/10.1007/978-3-031-56467-3_15.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, *5*(4), 537–550. https://doi.org/10.1109/72.298224.

Bucsuházy, K., Matuchová, E., Zúvala, R., Moravcová, P., Kostíková, M., & Mikulec, R. (2020). Human factors contributing to the road traffic accident occurrence. *Transportation Research Procedia*, *45*, 555–561. https://doi.org/10.1016/j.trpro.2020.03.057.

Cai, X., Gan, Y., Wu, M., & Wu, J. (2023). Weak supervised sound event detection based on puzzle CAM. *IEEE Access*, *11*, 89290–89297. https://doi.org/10.1109/ACCESS.2023.3305633.

Castorena, C., Cobos, M., Lopez-Ballester, J., & Ferri, F. J. (2024a). A safety-oriented framework for sound event detection in driving scenarios. *Applied Acoustics*, *215*, 109719. https://doi.org/10.1016/j.apacoust.2023.109719.

Castorena, C., Lopez-Ballester, J., De Rus, J. A., Cobos, M., & Ferri, F. J. (2025). Edge computing for driving safety: Evaluating deep learning models for cost-effective sound event detection. *The Journal of Supercomputing*, *81*(1), 1–15. https://doi.org/10.1007/s11227-024-06796-1.

Castorena, C., Lopez-Ballester, J., De Rus, J. A., Ferri, F. J., & Cobos, M. (2024b). Evaluation of real-time acoustic event detection models in driving scenarios. In *Proceedings of the 12th euro american conference on telematics and information systems* (pp. 1–4). https://doi.org/10.1145/3685243.3685264.

Chan, T. K., & Chin, C. S. (2022). Lightweight convolutional-iconformer for sound event detection. *IEEE Transactions on Artificial Intelligence*, *4*(4), 910–921. https://doi.org/10.1109/TAI.2022.3173582.

Chavdar, M., Kartalov, T., Ivanovski, Z., Taskovski, D., & Gerazov, B. (2023). Scarrie: A real-time system for sound event detection for assisted living. In *2023 30th international conference on systems, signals and image processing (IWSSIP)* (pp. 1–5). IEEE. https://doi.org/10.1109/IWSSIP58668.2023.10180246.

Chen, H.-T. M., & Thomas, M. (2020). Effects of audio learning on semi-autonomous driving safety. In *Proceedings of the human factors and ergonomics society annual meeting* (pp. 1181–1185). SAGE Publications Sage CA: Los Angeles, CA (*vol. 64*). Issue 1. https://doi.org/10.1177/1071181320641282.

Chen, Z., & Huang, Q. (2024). Sound event localization and detection using parallel multi-attention enhancement. *Circuits, Systems, and Signal Processing*, *43*(1), 545–567. https://doi.org/10.1007/s00034-023-02489-x.

Crundall, D., Van Loon, E., Baguley, T., & Kroll, V. (2021). A novel driving assessment combining hazard perception, hazard prediction and theory questions. *Accident Analysis & Prevention*, *149*, 105847. https://doi.org/10.1016/j.aap.2020.105847.

Dao, T., Guo, M., & Ma, M. (2024). Sound event localization and detection using a spatial omni-dimensional dynamic interactions network. *Signal, Image and Video Processing*, *18*(2), 1911–1917. https://doi.org/10.1007/s11760-023-02901-8.

Darbellay, G. A., & Vajda, I. (1999). Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, *45*(4), 1315–1321. https://doi.org/10.1109/18.761290.

Dewing, O. (2022). How spatial audio can improve the safety of self-driving cars. https://digitalcommons.lmu.edu/honors-research-and-exhibition/2022fall/Dec12/2/.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(02), 185–205. https://doi.org/10.1142/S0219720005001004.

Diwakar, M. P., & Gupta, B. (2024). Vggish deep learning model: Audio feature extraction and analysis. In *International conference on data management, analytics & innovation* (pp. 59–70). Springer. https://doi.org/10.1007/978-981-97-3245-6_5.

Du, Y., & Ding, R. (2014). Detection of traffic sound for driving safety: Effects of car body sound insulation. *Noise Control Engineering Journal*, *62*(6), 436–448. https://doi.org/10.3397/1/376241.

El-Latif, E. I. A., El-Sayad, N. E., Mohammed, K. K., Darwish, A., & Hassanien, A. E. (2024). Vggish transfer learning model for the efficient detection of payload weight of drones using mel-spectrogram analysis. *Neural Computing and Applications*, *36*(21), 12883–12899. https://doi.org/10.1007/s00521-024-09661-7.

Gao, L., Mao, Q., & Dong, M. (2024). On local temporal embedding for semi-supervised sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, . https://doi.org/10.1109/TASLP.2024.3369529.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., & Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (pp. 776–780). IEEE. https://doi.org/10.1109/ICASSP.2017.7952261.

Gharahbagh, A. A., Hajihashemi, V., & Tavares, J. M. R. S. (2021). A k-means wavelet neuro-fuzzy method to forecast the number of road accidents: A case study of Karaj-Qazvin highway. In *16th doctoral symposium in informatics engineering* (p. 112).

Gong, Y., Chung, Y.-A., & Glass, J. (2021). AST: Audio spectrogram transformer. https://doi.org/10.48550/arXiv.2104.01778.

Group, M. T. (2023). Freesound. https://freesound.org/home/.

Habibzadeh, Y., Yarmohammadian, M. H., & Sadeghi-Bazargani, H. (2023). Driving hazard perception components: A systematic review and meta-analysis. *Bulletin of Emergency & Trauma*, *11*(1), 1. https://doi.org/10.30476/beat.2023.95410.1356.

Hajihashemi, V., Alavigharahbagh, A., Machado, J. J. M., & Tavares, J. M. R. S. (2025a). Novel sound event and sound activity detection framework based on intrinsic mode functions and deep learning. *Multimedia Tools and Applications*, *84*(14), 13515–13543. https://doi.org/10.1007/s11042-024-19557-2.

Hajihashemi, V., Alavigharahbagh, A., Oliveira, H. S., Cruz, P. M., & Tavares, J. M. R. S. (2021). Novel time-frequency based scheme for detecting sound events from sound background in audio segments. In *Ibero American congress on pattern recognition* (pp. 402–416). Springer. https://doi.org/10.1007/978-3-030-93420-0_38.

Hajihashemi, V., Campos Ferreira, M., Machado, J. J. M., & Tavares, J. M. R. S. (2025b). Deep learning-based acoustic event detection and classification using cochleogram images. In *Iberian conference on information systems and technologies* (pp. 95–107). Springer. https://doi.org/10.1007/978-3-032-10728-2_9.

Hajihashemi, V., Gharahbagh, A. A., Cruz, P. M., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. R. S. (2022). Binaural acoustic scene classification using wavelet scattering, parallel ensemble classifiers and nonlinear fusion. *Sensors*, *22*(4), 1535. https://doi.org/10.3390/s22041535.

Hajihashemi, V., Gharahbagh, A. A., Machado, J. J. M., & Tavares, J. M. R. S. (2023). Audio event detection based on cross correlation in selected frequency bands of spectrogram. In *World conference on information systems and technologies* (pp. 182–191). Springer. https://doi.org/10.1007/978-3-031-45651-0_19.

Hajihashemi, V., Gharahbagh, A. A., Machado, J. J. M., & Tavares, J. M. R. S. (2026). Mel-based feature extraction for acoustic event detection and classification: A systematic review and research roadmap. *Expert Systems with Applications*, (132549). https://doi.org/10.1016/j.eswa.2026.132549.

Hebbar, R., Bose, D., Somandepalli, K., Vijai, V., & Narayanan, S. (2023). A dataset for audio-visual sound event detection in movies. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10094781.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B. et al. (2017). CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 131–135). IEEE. https://doi.org/10.1109/ICASSP.2017.7952132.

Howard, A. G. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861.

Imoto, K., Mishima, S., Arai, Y., & Kondo, R. (2022). Impact of data imbalance caused by inactive frames and difference in sound duration on sound event detection performance. *Applied Acoustics*, *196*, 108882. https://doi.org/10.1016/j.apacoust.2022.108882.

Jayalakshmi, S. L., Chandrakala, S., & Nedunchelian, R. (2018). Global statistical features-based approach for acoustic event detection. *Applied Acoustics*, *139*, 113–118. https://doi.org/10.1016/j.apacoust.2018.04.026.

Jocher, G. (2020). Ultralytics/yolov5: v3.1 - bug fixes and performance improvements. https://doi.org/10.5281/zenodo.4154370

Khandelwal, T., Das, R. K., Koh, A., & Chng, E. S. (2023). Leveraging audio-tagging assisted sound event detection using weakified strong labels and frequency dynamic convolutions. In *2023 IEEE Statistical signal processing workshop (SSP)* (pp. 329–333). IEEE. https://doi.org/10.1109/SSP53291.2023.10208013.

Kim, D., Park, S., Han, D. K., & Ko, H. (2021). Multi-band CNN architecture using adaptive frequency filter for acoustic event classification. *Applied Acoustics*, *172*, 107579. https://doi.org/10.1016/j.apacoust.2020.107579.

Li, K., Song, Y., Dai, L.-R., McLoughlin, I., Fang, X., & Liu, L. (2023). AST-SED: An effective sound event detection method based on audio spectrogram transformer. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10096853.

Li, K., Yang, S., Zhao, L., & Wang, W. (2024). Weakly labeled sound event detection with a capsule-transformer model. *Digital Signal Processing*, *146*, 104347. https://doi.org/10.1016/j.dsp.2023.104347.

Liang, Y., Long, Y., Li, Y., Liang, J., & Wang, Y. (2022). Joint framework with deep feature distillation and adaptive focal loss for weakly supervised audio tagging and acoustic event detection. *Digital Signal Processing*, *123*, 103446. https://doi.org/10.1016/j.dsp.2022.103446.

Liu, C.-C., Chen, C.-P., Lu, C.-L., Chan, B.-C., Cheng, Y.-H., Chuang, H.-F., & Chen, W.-Y. (2023). Regression-based sound event detection with semi-supervised learning. In *2023 Asia pacific signal and information processing association annual summit and conference (APSIPA ASC)* (pp. 2336–2342). IEEE. https://doi.org/10.1109/APSIPAASC58517.2023.10317136.

Liu, Y., Zhu, W., Dong, L., Zhang, Y., & Guo, X. (2025). Enhancing interpretability in video-based personality trait recognition using SHAP analysis. *Multimedia Systems*, *31*(2), 1–21. https://doi.org/10.1007/s00530-025-01690-z.

Ludena-Choez, J., & Gallardo-Antolin, A. (2016). Acoustic event classification using spectral band selection and non-negative matrix factorization-based features. *Expert Systems with Applications*, *46*, 77–86. https://doi.org/10.1016/j.eswa.2015.10.018.

Luo, S., Feng, Y., Liu, Z. J., Ling, Y., Dong, S., & Ferry, B. (2023). High precision sound event detection based on transfer learning using transposed convolutions and feature pyramid network. In *2023 IEEE International conference on consumer electronics (ICCE)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCE56470.2023.10043383.

MagicHub (2023). ASR-CabNois: A cabin noise dataset. https://magichub.com/datasets/in-vehicle-noise-dataset/.

Malmberg, C. (2021). Real-time audio classification on an edge device: Using YAMNet and TensorFlow lite. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1605037&dswid=1114.

Mandal, G., De, P., & Bhattacharya, D. (2021). Real-time fast fog removal approach for assisting drivers during dense fog on hilly roads. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9877–9889. https://doi.org/10.1007/s12652-020-02734-0.

Martín-Morató, I., Harju, M., Ahokas, P., & Mesaros, A. (2023). Training sound event detection with soft labels from crowdsourced annotations. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10095504.

Meng, J., Wang, X., Wang, J., Teng, X., & Xu, Y. (2022). A capsule network with pixel-based attention and BGRU for sound event detection. *Digital Signal Processing*, *123*, 103434. https://doi.org/10.1016/j.dsp.2022.103434.

Mesaros, A., Heittola, T., Benetos, E., Foster, P., Lagrange, M., Virtanen, T., & Plumbley, M. D. (2017). Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(2), 379–393. https://doi.org/10.1109/TASLP.2017.2778423.

Mesaros, A., Heittola, T., & Virtanen, T. (2016). Metrics for polyphonic sound event detection. *Applied Sciences*, *6*(6), 162. https://doi.org/10.3390/app6060162.

Mnasri, Z., Rovetta, S., & Masulli, F. (2023). Anomalous sound event detection based on one-class classification using variational autoencoders and interval type-2 fuzzy sets. In *2023 31st european signal processing conference (EUSIPCO)* (pp. 171–175). IEEE. https://doi.org/10.23919/EUSIPCO58844.2023.10290085.

Mondal, S., & Barman, A. D. (2022). Deep learning technique based real-time audio event detection experiment in a distributed system architecture. *Computers and Electrical Engineering*, *102*, 108252. https://doi.org/10.1016/j.compeleceng.2022.108252.

Moradi, H. (2025). Road and transport safety. https://transportsafety.ir/.

Mulimani, M., & Koolagudi, S. G. (2019). Robust acoustic event classification using fusion fisher vector features. *Applied Acoustics*, *155*, 130–138. https://doi.org/10.1016/j.apacoust.2019.05.020.

Nagatomo, K., Yasuda, M., Yatabe, K., Saito, S., & Oikawa, Y. (2022). On-line sound event localization and detection for real-time recognition of surrounding environment. *Applied Acoustics*, *199*, 108961. https://doi.org/10.1016/j.apacoust.2022.108961.

Organization, W. H. (2023). Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159.

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y., & Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/JSTSP.2019.2908700.

Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 1041–1044). https://doi.org/10.1145/2647868.2655045.

Salamon, J., MacConnell, D., Cartwright, M., Li, P., & Bello, J. P. (2017). Scaper: A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on applications of signal processing to audio and acoustics (WASPAA)* (pp. 344–348). IEEE. https://doi.org/10.1109/WASPAA.2017.8170052.

Serizel, R., Cornell, S., & Turpault, N. (2023). Performance above all? energy consumption vs. performance, a study on sound event detection with heterogeneous data. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10095938.

Serizel, R., Turpault, N., Shah, A., & Salamon, J. (2020). Sound event detection in synthetic domestic environments. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 86–90). IEEE. https://doi.org/10.1109/ICASSP40776.2020.9054478.

Shin, Y., Kim, Y. G., Choi, C.-H., Kim, D.-J., & Chun, C. (2023). Seld U-Net: Joint optimization of sound event localization and detection with noise reduction. *IEEE Access*, *11*, 105379–105393. https://doi.org/10.1109/ACCESS.2023.3318322.

Sun, Y., Li, J., Wang, L., Xv, J., & Liu, Y. (2024). Deep learning-based drone acoustic event detection system for microphone arrays. *Multimedia Tools and Applications*, *83*(16), 47865–47887. https://doi.org/10.1007/s11042-023-17477-1.

Svatos, J., & Holub, J. (2023). Impulse acoustic event detection, classification, and localization system. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–15. https://doi.org/10.1109/TIM.2023.3252631.

Tian, M., Li, R., & An, W. (2023). Sound event detection based on mel spectral envelope estimation and regression detection. In *2023 IEEE International conference on signal processing, communications and computing (ICSPCC)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICSPCC59353.2023.10400273.

Torosyan, N. (2019). Old phone ringtones as MIDI. https://www.kaggle.com/datasets/narektorosyan/old-phone-ringtones-as-midi.

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, *10*(5), 293–302. https://doi.org/10.1109/TSA.2002.800560.

Valani, B. (2023). Donate-a-cry-corpus-features-dataset. https://www.kaggle.com/datasets/bhoomikavalani/donateacrycorpusfeaturesdataset.

Wan, S., He, J., Wu, X., Ye, Y., Qin, P., & Fang, Z. (2025). A surrogate approach for real-time stress assessment of urban drivers using driving video data. *IEEE Transactions on Intelligent Transportation Systems*, . https://doi.org/10.1109/TITS.2025.3538647.

Wang, J., Yao, P., Deng, F., Tan, J., Song, C., & Wang, X. (2023a). NAS-DYMC: NAS-based dynamic multi-scale convolutional neural network for sound event detection. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10096621.

Wang, M., Li, Y., & Hu, Y. (2023b). Improved self-consistency training with selective feature fusion for sound event detection. In *2023 6th international conference on information communication and signal processing (ICICSP)* (pp. 460–464). IEEE. https://doi.org/10.1109/ICICSP59554.2023.10390568.

Wang, Q., Du, J., Wu, H.-X., Pan, J., Ma, F., & Lee, C.-H. (2023c). A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *31*, 1251–1264. https://doi.org/10.1109/TASLP.2023.3256088.

Wang, Y., & Lu, Y. (2024). Yolov10-based object detection method with optimized defogging algorithm for autonomous driving in foggy environments. In *2024 International conference on intelligent robotics and automatic control (IRAC)* (pp. 391–396). IEEE. https://doi.org/10.1109/IRAC63143.2024.10871902.

Wang, Y., Zhao, G., Xiong, K., & Shi, G. (2022). MSFF-NET: Multi-scale feature fusing networks with dilated mixed convolution and cascaded parallel framework for sound event detection. *Digital Signal Processing*, *122*, 103319. https://doi.org/10.1016/j.dsp.2021.103319.

Wei, J., Zhang, Q., & Ning, W. (2023). Self-supervised learning representation for abnormal acoustic event detection based on attentional contrastive learning. *Digital Signal Processing*, *142*, 104199. https://doi.org/10.1016/j.dsp.2023.104199.

Wilkins, J., Seetharaman, P., Wahl, A., & Pardo, B. (2018). VocalSet: A singing voice dataset. In *Ismir* (pp. 468–474). https://doi.org/10.5281/zenodo.1193957.

Wu, S., Wang, Y., Hu, Z., & Liu, J. (2023). HAAC: Hierarchical audio augmentation chain for accdoa described sound event localization and detection. *Applied Acoustics*, *211*, 109541. https://doi.org/10.1016/j.apacoust.2023.109541.

Xiao, S., Zhang, X., & Zhang, P. (2023). Multi-dimensional frequency dynamic convolution with confident mean teacher for sound event detection. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10096306.

Xin, Y., Yang, D., Cui, F., Wang, Y., & Zou, Y. (2023). Improving weakly supervised sound event detection with causal intervention. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10096245.

Xu, L., Wang, L., Bi, S., Liu, H., & Wang, J. (2023). Semi-supervised sound event detection with pre-trained model. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10095687.

Xue, L., Liu, H., Zhou, Y., & Gan, L. (2023). RESNet-conformer network using multi-scale channel attention for sound event localization and detection in real scenes. In *2023 International conference on wireless communications and signal processing (WCSP)* (pp. 25–30). IEEE. https://doi.org/10.1109/WCSP58612.2023.10404673.

Yang, H., Luo, L., Wang, M., Song, X., & Mi, F. (2023). Sound event detection using multi-scale dense convolutional recurrent neural network with lightweight attention. In *2023 3rd international conference on electronic information engineering and computer (EIECT)* (pp. 35–40). IEEE. https://doi.org/10.1109/EIECT60552.2023.10442997.

Ye, J., Kobayashi, T., & Murakawa, M. (2017). Urban sound event classification based on local and global features aggregation. *Applied Acoustics*, *117*, 246–256. https://doi.org/10.1016/j.apacoust.2016.08.002.

Yuan, Z., Wu, D., Wu, X., & Qu, T. (2023). Sound event localization and detection based on iterative separation in embedding space. In *2023 6th international conference on information communication and signal processing (ICICSP)* (pp. 455–459). IEEE. https://doi.org/10.1109/ICICSP59554.2023.10390682.

Zhang, H., Wu, M., Cai, X., & Liu, W. (2024). An event-scene co-operative analysis network with dual-stream attention convolution module and soft parameter-sharing. *Applied Acoustics*, *222*, 110066. https://doi.org/10.1016/j.apacoust.2024.110066.

Zhang, Y., Zhang, K., Wang, J., & Su, Y. (2021). Robust acoustic event recognition using AVMD-PWVD time-frequency image. *Applied Acoustics*, *178*, 107970. https://doi.org/10.1016/j.apacoust.2021.107970.

Zhao, D., Ding, K., Qi, X., Chen, Y., & Feng, H. (2023). Sound event localization and detection based on deep learning. *Journal of Systems Engineering and Electronics*, *35*(2), 294–301. https://doi.org/10.23919/JSEE.2023.000110.

Zhao, F., Li, R., Liu, X., & Xu, L. (2022). Soft-median selection: An adaptive feature smoothening method for sound event detection. *Applied Acoustics*, *192*, 108715. https://doi.org/10.1016/j.apacoust.2022.108715.

Zhao, Q., Yang, L., & Lyu, N. (2024). A driver stress detection model via data augmentation based on deep convolutional recurrent neural network. *Expert Systems with Applications*, *238*, 122056. https://doi.org/10.1016/j.eswa.2023.122056.

Zhou, Y., & Wan, H. (2023). Dual-branch attention module-based network with parameter sharing for joint sound event detection and localization. *EURASIP Journal on Audio, Speech, and Music Processing*, *2023*(1), 27. https://doi.org/10.1186/s13636-023-00292-9.