VNIVERSITAT DE VALÈNCIA

Universitat de València

Escola Tècnica Superior d'Enginyeria

Programa de Doctorat en Tecnologies de la Informació, Comunicacions i Computació

Contributions to Deep Learning-Based Sound Event Classification and Detection for Driving Environments

Doctoral Thesis

Author:

Carlos Mauricio Castorena Lara

Thesis supervisors:

Francesc J. Ferri and Maximo Cobos

June 2025

Agradecimientos

Quiero agradecer a todas las personas que me han acompañado y apoyado durante mis estudios de doctorado hasta la finalización de esta tesis.

En primer lugar, a Karem López, por todo su apoyo, cariño y paciencia en cada etapa de este proceso. Su motivación ha sido fundamental para superar los momentos difíciles y seguir adelante con ilusión.

A mi familia, que, pese a la distancia, siempre me ha brindado su apoyo incondicional.

A mis profesores Máximo Cobos y Francesc Ferri, por su guía experta, sus consejos enriquecedores y por compartir conmigo su conocimiento y entusiasmo. Su dedicación ha sido clave para mi formación académica y personal.

También quiero agradecer a mis amigos y compañeros, tanto de la Universitat de València como de otras universidades, con quienes he tenido el placer de compartir experiencias, proyectos y momentos inolvidables. Su colaboración ha enriquecido esta etapa.

A todos, gracias por formar parte de este logro. Esta tesis es también fruto de toda su confianza y apoyo constante.

Prefacio

Esta tesis, con distinción internacional, ha sido posible gracias a la generosa financiación proporcionada por la Generalitat Valenciana a través del programa Santiago Grisolía (GRISOLIAP/2021/060, CPI-21-232), así como a una beca para estancias predoctorales de investigación fuera de la Comunitat Valenciana (CIBEFP/2023/063). Estas contribuciones han sido esenciales para el desarrollo de este trabajo y la consecución de su distinción internacional.

Este documento cumple con todas las normativas y requisitos académicos vigentes, que incluyen:

- **Tesis presentada en un idioma no oficial:**
  - Este documento contiene un resumen (de al menos 5000 palabras) en un idioma oficial, que en este caso es el español. Dicho resumen se encuentra en el Apéndice E.
- **Distinción internacional:**
  - La presente tesis está redactada íntegramente en inglés, idioma comúnmente utilizado en la comunidad científica.
  - Se realizó una estancia breve de investigación fuera de España, con una duración mínima de tres meses. En el contexto de esta tesis, dicha estancia tuvo lugar en la Universidad de Tampere bajo la supervisión de Tuomas Virtanen, dentro del grupo de investigación en Procesamiento de Señales, con una duración total de cuatro meses.

Resumen

Esta tesis aborda el reconocimiento auditivo en entornos vehiculares como una vía complementaria y robusta para mejorar la seguridad y la comprensión contextual en escenarios de conducción. A pesar del creciente interés en los sistemas avanzados de asistencia al conductor, el canal auditivo ha sido tradicionalmente subexplotado frente a modalidades como la visión por computadora o la fusión sensorial basada en señales del vehículo. Este trabajo propone un enfoque sistemático para integrar la percepción sonora en vehículos inteligentes, abordando tres contribuciones principales: la caracterización y modelado de eventos sonoros distractores, el diseño de un sistema eficiente de detección basado en YOLO adaptado al dominio acústico, y una estrategia de adaptación incremental para mejorar la personalización y robustez en condiciones reales.

La primera contribución se centra en la construcción de una taxonomía de eventos distractores de naturaleza acústica, basada en una revisión bibliográfica y un análisis empírico de contextos de conducción. A partir de esta taxonomía, se desarrollan conjuntos de datos sintéticos y reales que permiten entrenar modelos con control preciso sobre variables críticas como la densidad sonora, el tipo de evento, la duración o la superposición de fuentes. Esta base experimental facilita el análisis de métricas robustas de detección, permitiendo una evaluación realista de la capacidad de los modelos en escenarios complejos.

La segunda contribución introduce una adaptación de la arquitectura YOLO, concebida originalmente para visión por computadora, al dominio de detección de eventos sonoros. La arquitectura propuesta prescinde de módulos recurrentes y se compone exclusivamente de capas convolucionales, resultando en una mejora sustancial en eficiencia computacional y latencia, características esenciales para aplicaciones en tiempo. Los experimentos demuestran que el modelo logra una detección precisa de eventos distractores en escenarios sintéticos multiclase, manteniendo una baja tasa de falsos positivos incluso en entornos acústicamente saturados. Aunque el rendimiento disminuye en condiciones reales —debido a reverberación no modelada, interferencias imprevistas o ruido de fondo

cambiante—, el sistema mantiene una utilidad práctica significativa, validando su potencial como módulo de asistencia al conductor en situaciones reales.

La tercera contribución aborda el desafío de la brecha entre dominios, proponiendo una estrategia semisupervisada de selección y adaptación incremental (ISA). Esta técnica selecciona de forma informada un pequeño subconjunto de muestras representativas dentro de un conjunto no etiquetado, utilizando un espacio latente generado por el modelo y una técnica de agrupamiento modificada que garantiza la representatividad global. Las muestras seleccionadas son luego etiquetadas por el usuario y empleadas para adaptar el modelo, minimizando la intervención requerida. Con menos de 25 muestras etiquetadas, el modelo consigue mejoras significativas en tareas como el reconocimiento de emociones, abriendo la puerta a su extensión hacia otras tareas auditivas, como la detección personalizada de eventos distractores. La estrategia ofrece un equilibrio favorable entre costo de etiquetado y ganancia de rendimiento, facilitando implementaciones prácticas y escalables.

Las aplicaciones prácticas de estas contribuciones son amplias. La integración de detección auditiva en vehículos puede mejorar los sistemas de monitoreo del conductor, detectar fuentes de distracción internas y externas, y permitir una personalización eficiente basada en condiciones acústicas o usuarios específicos. La posibilidad de implementar modelos eficientes en hardware embebido sin depender de la nube amplía el alcance hacia vehículos comerciales, flotas, taxis autónomos o contextos con recursos limitados. Además, los recursos generados —como conjuntos de datos revisados y taxonomías propuestas— aportan valor inmediato para la comunidad investigadora e industrial, facilitando comparaciones, replicabilidad y nuevas líneas de investigación.

Finalmente, se delimitan líneas futuras que incluyen aumentar el realismo en los datos sintéticos, la expansión dinámica de taxonomías multimodales y el desarrollo de modelos que integren conocimiento preentrenado o se adapten a etiquetado débil. En conjunto, esta tesis contribuye sustancialmente a la construcción de sistemas auditivos vehiculares más inteligentes, eficientes y adaptables, sentando las bases para una movilidad más segura, sensible al contexto y centrada en el usuario.

Resum

Aquesta tesi aborda el reconeixement auditiu en entorns vehiculars com una via complementària i robusta per millorar la seguretat i la comprensió contextual en escenaris de conducció. Tot i l'interès creixent pels sistemes avançats d'assistència al conductor, el canal auditiu ha estat tradicionalment subexplotat respecte a modalitats com la visió per ordinador o la fusió sensorial basada en senyals del vehicle. Aquest treball proposa un enfocament sistemàtic per integrar la percepció sonora en vehicles intel·ligents, abordant tres contribucions principals: la caracterització i modelatge d'esdeveniments sonors distractors, el disseny d'un sistema eficient de detecció basat en YOLO adaptat al domini acústic, i una estratègia d'adaptació incremental per millorar la personalització i la robustesa en condicions reals.

La primera contribució se centra en la construcció d'una taxonomia d'esdeveniments distractors acústics, basada en una revisió bibliogràfica i una anàlisi empírica de contextos de conducció. A partir d'aquesta taxonomia, es desenvolupen conjunts de dades sintètiques i reals que permeten entrenar models amb control precís sobre variables crítiques com la densitat sonora, el tipus d'esdeveniment, la duració o la superposició de fonts. Aquesta base experimental facilita l'anàlisi de mètriques robustes de detecció, permetent una avaluació realista de la capacitat dels models en escenaris complexos.

La segona contribució presenta una adaptació de l'arquitectura YOLO, originalment concebuda per a la visió per ordinador, al domini de detecció d'esdeveniments sonors. L'arquitectura proposada prescindeix de mòduls recurrents i es compon exclusivament de capes convolucionals, cosa que implica una millora substancial en eficiència computacional i latència, característiques fonamentals per a aplicacions en temps real a bord dels vehicles. Els experiments mostren que el model aconsegueix una detecció precisa d'esdeveniments distractors en escenaris sintètics multiclasse, mantenint una baixa taxa de falsos positius fins i tot en entorns acústicament saturats. Encara que el rendiment disminueix en condicions reals —degut a factors com reverberacions no modelades,

interferències imprevistes o soroll de fons variable—, el sistema manté una utilitat pràctica significativa, validant el seu potencial com a mòdul d'assistència al conductor en situacions del món real.

La tercera contribució aborda el repte de la bretxa entre dominis, proposant una estratègia semisupervisada de selecció i adaptació incremental (ISA). Aquesta tècnica selecciona de manera informada un petit subconjunt de mostres representatives dins d'un conjunt no etiquetat, utilitzant un espai latent generat pel model i una tècnica d'agrupament modificada que garanteix la representativitat en tot l'espai. Les mostres seleccionades són després etiquetades per l'usuari i emprades per adaptar el model, minimitzant la intervenció requerida. Amb menys de 25 mostres etiquetades, el model obté millores significatives en tasques com el reconeixement d'emocions, obrint la porta a la seua extensió a altres tasques auditives, com la detecció personalitzada d'esdeveniments distractors. L'estratègia ofereix un bon equilibri entre el cost d'etiquetatge i el guany de rendiment, permetent implementacions pràctiques i escalables.

Les aplicacions pràctiques d'aquestes contribucions són àmplies. La integració de detecció auditiva en vehicles pot millorar els sistemes de monitoratge del conductor, detectar fonts de distracció internes i externes, i permetre una personalització eficient basada en condicions acústiques o usuaris específics. La possibilitat d'implementar models eficients en maquinari embegut sense dependre del núvol amplia l'abast cap a vehicles comercials, flotes, taxis autònoms o contextos amb recursos limitats. A més, els recursos generats —com els conjunts de dades revisades i les taxonomies proposades— ofereixen un valor immediat per a la comunitat investigadora i industrial, facilitant comparacions, la replicabilitat i noves línies d'investigació.

Finalment, es delimiten línies futures que inclouen l'augment del realisme en les dades sintètiques, l'expansió dinàmica de taxonomies multimodals i el desenvolupament de models que integren coneixement preentrenat o que s'adapten a etiquetatge dèbil. En conjunt, aquesta tesi contribueix substancialment a la construcció de sistemes auditius vehiculars més intel·ligents, eficients i adaptables, assentant les bases per a una mobilitat més segura, sensible al context i centrada en l'usuari.

Abstract

This thesis addresses auditory recognition in vehicular environments as a complementary and robust approach to improving safety and contextual understanding in driving scenarios. Despite the growing interest in advanced driver assistance systems, the auditory channel has traditionally been underutilized compared to modalities such as computer vision or sensor fusion based on vehicle signals. This work proposes a systematic approach to integrating sound perception into intelligent vehicles, focusing on three main contributions: the characterization and modeling of distracting auditory events, the design of an efficient detection system based on YOLO adapted to the acoustic domain, and an incremental adaptation strategy to enhance personalization and robustness under real-world conditions.

The first contribution focuses on the construction of a taxonomy of distracting acoustic events, based on a literature review and empirical analysis of driving contexts. From this taxonomy, both synthetic and real datasets are developed, enabling model training with precise control over critical variables such as sound density, event type, duration, and source overlap. This experimental foundation supports the analysis of robust detection metrics, allowing a realistic evaluation of model performance in complex scenarios.

The second contribution introduces an adaptation of the YOLO architecture—originally designed for computer vision—to the task of detecting sound events. The proposed model eliminates recurrent modules and is composed exclusively of convolutional layers, resulting in substantial improvements in computational efficiency and latency, which are key for real-time onboard applications. Experimental results show that the model achieves accurate detection of distracting events in synthetic multiclass scenarios while maintaining a low false positive rate even in acoustically saturated environments. Although performance decreases under real-world conditions—due to factors such as unmodeled reverberation, unexpected interferences, or changing background noise—the system retains significant practical utility, validating its potential as

an assistive module for real-life driving contexts.

The third contribution tackles the domain gap challenge by proposing a semi-supervised strategy for selection and incremental adaptation (ISA). This technique intelligently selects a small subset of representative samples from an unlabeled dataset, using a latent space generated by the model and a modified clustering technique that ensures representativeness across the space. Selected samples are then labeled by the user and used to adapt the model, minimizing the required intervention. With fewer than 25 labeled samples, the model shows significant improvements in tasks such as emotion recognition, opening the door to extending the strategy to other auditory tasks like personalized detection of distracting events. This method offers a favorable balance between labeling cost and performance gain, enabling scalable and practical deployments.

The practical applications of these contributions are broad. Integrating auditory detection into vehicles can enhance driver monitoring systems, identify both internal and external distraction sources, and enable efficient personalization based on specific acoustic conditions or user profiles. The possibility of deploying efficient models on embedded hardware without cloud dependency expands the applicability to commercial vehicles, fleets, autonomous taxis, or low-resource settings. Additionally, the resources generated—such as curated datasets and proposed taxonomies—offer immediate value to the research and industrial community, supporting benchmarking, replicability, and future studies.

Finally, future work includes improving the realism of synthetic data, dynamically expanding multimodal taxonomies, and developing models that incorporate pretrained knowledge or adapt to weakly labeled data. Altogether, this thesis makes a substantial contribution toward the development of smarter, more efficient, and adaptable auditory systems for vehicles, laying the groundwork for safer, context-aware, and user-centered mobility.

Contents

List of Figures

List of Tables

Symbols and Acronyms Glossary

Terms	Description	Terms	Description

General Acronyms

DCASE	Detection and classification of Acacoustic scenes and events	DTF	Discrete Fourier transform
ESD	Emotional speech dataset	IEMOCAP	Interactive emotional dyadic motion capture database
ISA	Incremental selection and adaptation	MESD	Mexican emotional speech database
MSP	Multimodal signal processing corpus	ONNX	Open neural network exchange
PDASH	Protodash cell content using parbox	RAVDESS	Ryerson audio-visual database of emotional speech and song
SD	Source domain	SEC	Sound event classification
SED	Sound event detection	SER	Speech emotion recognition
STFT	Short-time Fourier transform	SS	Selected samples
TD	Target domain		

Networks and Components

Bi-GRU	Bidirectional gated recurrent unit	CNN	Convolutional neural network
CRNN	Convolutional recurrent neural network	EMA	Exponential moving average
FPN	Feature pyramid network	GLU	Gated linear unit
GRU	Gated recurrent unit	LSTM	Long Short-Term Memory
ReLU	Rectified linear unit	RNN	Recurrent neural network
YOLO	You only hear once	YOHO	You only look once
$\mathbf{X}$, $\mathbf{x}$	Raw audio, mini-bash and single sample	$\hat{\mathbf{X}}, \hat{\mathbf{x}}$	Latent feature representation, mini-bash and single sample
$\mathbf{Y}$, $\mathbf{y}$	Label, mini-bash and single sample	$\hat{\mathbf{Y}}, \hat{\mathbf{y}}$	Prediction, mini-bash and single sample
$\mathcal{L}$	Generic loss function	$\mathcal{L}_{\text{BCE}}$	Binary cross-entropy loss function
$\mathcal{L}_{\text{C}}$	Center loss function	$\mathcal{L}_{\text{CCE}}$	Categorical cross-entropy loss function
$\mathcal{L}_{\text{CIoU}}$	Center intersection over union loss function	$\mathcal{L}_{\text{MSE}}$	Mean squared error loss function

Terms	Description	Terms	Description

Metrics

ACC	Accuracy	AUC	Area under the curve
CTTC	Cross-trigger tolerance criterion	DTC	Detection tolerance criterion
F1	F1-score	FN	False negatives
FP	False positives	GTC	Ground truth intersection criterion
G-mean	Geometric mean	PSDS	Polyphonic sound detection score
TN	True negatives	TP	True positives
θ	Threshold		

Datasets Nomenclature

`SED_SYN_TRAIN`	Synthetic training set for SED	`SED_SYN_TEST`	Synthetic test set for SED
`SED_REAL_S1`	Real subset scenario 1 for SED	`SED_REAL_S2`	Real subset scenario 2 for SED
`SED_REAL_S3`	Real subset scenario 3 for SED	`SER_SD_TRAIN`	Source domain train set for SER
`SER_SD_TEST`	Source domain test set for SER	`SER_TD_TRAIN`	Target domain train set for SER
`SER_TD_TEST`	Target domain test set for SER		

Chapter 1:
Introduction

1.1 Background

Automatic sound event classification and detection play a crucial role in a wide range of real-world applications, from wildlife monitoring to industrial safety and smart home automation [1, 2]. By capturing and analyzing sound events, these systems provide valuable information that enhances environmental understanding, detects anomalies, and facilitates decision-making. Thanks to rapid advances in the field of artificial intelligence, it is now possible to develop increasingly accurate and faster solutions, enabling their implementation in more and more real-world applications.

One of the most critical applications is in driving environments [3, 4, 5, 6], where sound is an essential source of information for both human drivers and intelligent vehicle systems. Modern vehicles are increasingly equipped with sophisticated audio technologies, including virtual assistants that process voice commands to control systems such as climate control, navigation, and entertainment [7, 8]. Moreover, they can play a key role in road safety by detecting sound events such as sirens, honks, and sudden braking, which can alert drivers to potential hazards [9]. The detection of sound events, in combination with other types of sensors and visual information, helps achieve a comprehensive understanding of the driving environment. Analyzing the nature and impact of these acoustic distractions from different sound sources can help design systems that mitigate their risks. Beyond safety-related applications, detecting critical sound events from both external sources (e.g., road noise and other vehicles) and internal sources (e.g., conversations, alerts), classification and detection systems can help assess the cognitive and emotional state of drivers [10, 11], which in turn can support informed decisions to reduce driving risk. Deep learning has become a powerful tool to address the challenges of classification

and detection. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based architectures have demonstrated remarkable capabilities in extracting meaningful representations from raw audio data. For example, CNNs excel at learning spatial patterns in spectrograms [12], RNNs capture temporal dependencies in sequential data [13], and transformer-based models have shown excellent performance in managing long-range dependencies in large datasets [14]. However, despite their success, these models face significant challenges when applied to driving scenarios. A key limitation is the scarcity of labeled datasets specifically designed for vehicular environments. While datasets like UrbanSound8K and others focused on home environments provide valuable benchmarks, they fail to fully capture the unique acoustic conditions found inside vehicles or on the road. Variations in background noise, cabin acoustics, and diverse driving scenarios complicate the implementation of robust models in real-world applications.

Another critical issue is the generalization of deep learning models to new and variable conditions. Models trained on controlled datasets often struggle when exposed to novel acoustic environments, where factors such as noise levels, capture conditions, and sound source characteristics can vary significantly. This limitation is particularly evident in tasks like speech emotion recognition (SER) [15], a specific classification problem where emotions must be recognized across different speakers and situational contexts. Similarly, sound event classification and detection models trained in controlled environments often experience performance degradation when deployed in real-world driving conditions, where sound events may overlap, be occluded by noise, or exhibit variations not present during training.

Moreover, practical applications require a precise balance between model accuracy and computational efficiency. Many real-time acoustic monitoring systems are implemented on embedded or edge computing devices with limited processing capacity. High-complexity deep learning models, while accurate, often require considerable computational resources, leading to latency issues that make them impractical for real-time decision-making in safety-critical applications [16]. Additionally, vehicle monitoring systems must operate under

strict power constraints, especially in electric or hybrid vehicles where energy efficiency is a priority. This balance between performance and efficiency remains a challenge, requiring optimization strategies to ensure real-time responsiveness while maintaining high accuracy.

1.2 Objectives

The main objective of this thesis is to advance the field of sound event classification and detection, particularly in complex and dynamic environments such as driving scenarios. This work aims to develop innovative deep learning models, frameworks, and methodologies that improve the efficiency, accuracy, and generalization capability of acoustic systems, enabling their deployment in real-world conditions. These advancements will focus on enhancing sound detection and classification systems not only in terms of precision but also in speed, ensuring that models can operate in real time and meet the demanding requirements of automotive applications under resource-constrained conditions.

A key objective is to design and implement a comprehensive framework for detecting and classifying driving-related sounds, including potential hazards, distractions, and other critical events that may affect driver safety. To achieve this, the study will thoroughly explore various sound sources encountered in a vehicle, both internal and external. This information will be consolidated into a taxonomy that connects these sound sources and analyzes their safety implications, comparing them with the current state of the art. Additionally, a specialized dataset will be developed, incorporating both real and synthetic data, to better train and validate models designed for the unique acoustic conditions present in driving environments.

In parallel, the research aims to optimize deep learning models for deployment on low-cost, resource-constrained devices, ensuring that sound detection and classification models can operate efficiently in real-time systems while maintaining high accuracy. These models will be tested to ensure effective performance within systems with limited computational resources, addressing the practical challenges of edge computing and on-device processing.

In addition, this thesis aims to improve the performance and generalization ability of acoustic models through domain adaptation. Specifically, the research will explore techniques for gradually improving generalization, applying them in a particular case: speech emotion recognition, given that each speaker expresses emotions in a unique way. This challenge is not limited to speech analysis alone but also addresses a broader problem that can arise in various contexts, such as changes in microphones, variations in environments, or general conditions. Therefore, the findings of this research will contribute to improving generalization in classification and acoustic detection by allowing the proposed methodologies to be applied in other contexts.

1.3 Main Contributions

This thesis presents multiple contributions to the field of sound event classification and detection, underpinned by several peer-reviewed publications, including conference presentations and journal articles. Among them, are some key works that form the foundation of this research [9, 16, 17]. The main thesis contributions are summarized below:

- **A Framework for Acoustic Monitoring in Driving Environments**
  This contribution introduces a novel taxonomy specifically designed for sound event detection in driving environments. While existing taxonomies often focus on other domains, this framework is tailored to the unique acoustic conditions found in vehicles and roadways. Based on this taxonomy, a comprehensive dataset is created, incorporating both synthetic and real-world data to reflect the diverse range of sounds encountered in driving scenarios. This dataset serves as a foundation for developing and evaluating robust models, with the aim of enhancing the accuracy and reliability of classification and detection in automotive safety applications.

- **CNN-Based Approach for Efficient Sound Event Detection**
  A novel and lightweight neural network architecture based on the YOLO (You only look once) methodology is presented, focusing on real-time

performance in resource-constrained environments. The key contribution of this model lies not only in its ability to achieve competitive accuracy but also in its efficiency, making it ideal for deployment in edge computing scenarios — a critical factor for automotive applications where latency and power consumption are major concerns.

- **Incremental Domain Adaptation Strategy for Sound Events**
  One of the persistent challenges in sound event classification and detection is maintaining performance across varying acoustic environments. This thesis presents a semi-supervised domain adaptation strategy that incrementally improves model generalization without requiring extensive labeled data from each new environment, in particular, in emotion recognition, adapting the model to new speakers. This contribution enhances model robustness, ensuring adaptability to diverse driving scenarios, where acoustic conditions may change unpredictably.

1.4 Thesis Structure

This thesis is organized into several chapters, each addressing a key aspect of sound event classification and detection in driving environments. The structure is as follows:

- **Chapter 1: Introduction**
  Presents the motivation behind the research, outlines the main challenges in sound event classification and detection for driving environments—discussed in the background section—and defines the objectives and contributions of the thesis. It also introduces the document's structure.
- **Chapter 2: Theoretical Foundations of Sound Event Classification and Detection**
  Reviews the fundamental concepts of sound event detection and deep learning methods, along with relevant literature.
- **Chapter 3: A Framework for Acoustic Monitoring in Driving Environments**

Introduces a safety-oriented framework for sound event detection in vehicles. Describes the construction of a structured dataset under various acoustic conditions and presents a taxonomy for relevant sound sources. The chapter also discusses the methodology and its implications for driver safety.

- **Chapter 4: CNN-Based Approach for Efficient Sound Event Detection**

  Details the development of a deep learning model designed for real-time sound event detection under resource constraints. Emphasis is placed on achieving a balance between accuracy and computational efficiency.

- **Chapter 5: Incremental Domain Adaptation Strategy for Sound Events**

  Explores techniques to improve model generalization across varying acoustic environments. Proposes an incremental semi-supervised domain adaptation method and evaluates its effectiveness in real-world settings.

- **Chapter 6: Conclusions and Future Work**

  Summarizes the thesis contributions and their significance to the field. Also outlines directions for future research.

Chapter 2:
Theoretical Foundations of Sound Event Classification and Detection

This chapter introduces the fundamental concepts underlying sound event classification (SEC) and sound event detection (SED), providing a conceptual framework that supports the advanced methods and experimental results developed in Chapters 3, 4, and 5.

The discussion begins by defining SEC and SED as two core tasks in audio signal processing. While SEC assigns a global label to an entire audio segment, SED identifies both the type of sound and its precise temporal boundaries. Despite sharing similar principles, these tasks require distinct approaches, as explored in Section 2.1.

Effective SEC and SED rely on transforming raw waveforms into structured representations such as spectrograms and Mel spectrograms. These transformations, detailed in Section 2.2.1, enhance signal structure and reduce noise to facilitate feature extraction.

Section 2.2.2 outlines the main strategies used to extract relevant features from these representations. Local features capture time-frequency patterns, temporal features encode sequential progression, and transformer-based features offer high-level abstractions that support model performance. These features are then processed by classification and detection layers, introduced in Section 2.2.3, which convert the extracted information into predictions.

Training is performed using gradient-based optimization over labeled datasets. Section 2.2.4 explains relevant aspects of this process, including loss functions and regularization techniques. Model performance is assessed using standard evaluation metrics, as discussed in Section 2.2.5, which are essential for measuring the accuracy and robustness of sound event models.

Finally, Section 2.3 examines real-world limitations such as data imbalance,

domain variability, and computational constraints. Techniques such as data augmentation, model regularization, quantization, and pruning are reviewed as practical solutions for improving generalization and efficiency in real deployment scenarios.

2.1 Sound Event Classification and Detection

The tasks of SEC and SED are highly related to each other, with applications ranging from environmental monitoring and security systems [1, 18, 9] to intelligent assistance technologies [19, 20]. Both involve analyzing and categorizing sound signals, but while SEC focuses on assigning labels to entire audio segments, SED requires detecting individual sound events and their precise temporal locations. Despite these differences, both tasks share common theoretical foundations and methodological approaches.

SEC assigns a single label to a fixed-length audio segment, identifying the most predominant sound. This approach is widely used in music genre classification [21], speaker recognition [22], and acoustic scene analysis [23]. In some cases, it extends to multi-label classification [24, 25], where multiple labels are assigned to a single segment to account for polyphony, a phenomenon in which multiple sound events occur simultaneously [26, 27].

In contrast, SED requires not only identifying the sound events present in an audio stream but also determining their onset and offset times [28]. This additional temporal dimension makes SED significantly more complex, especially in real-world environments where multiple sounds overlap or occur in rapid succession. The ability to precisely localize sound events is critical in applications such as surveillance [29], distractor detection systems [16], music recognition [30], and medical monitoring [31], where the timing of an event directly impacts decision-making.

A key distinction between SEC and SED lies in the nature of their labels. SEC typically relies on weak labels, which indicate the presence of a specific sound

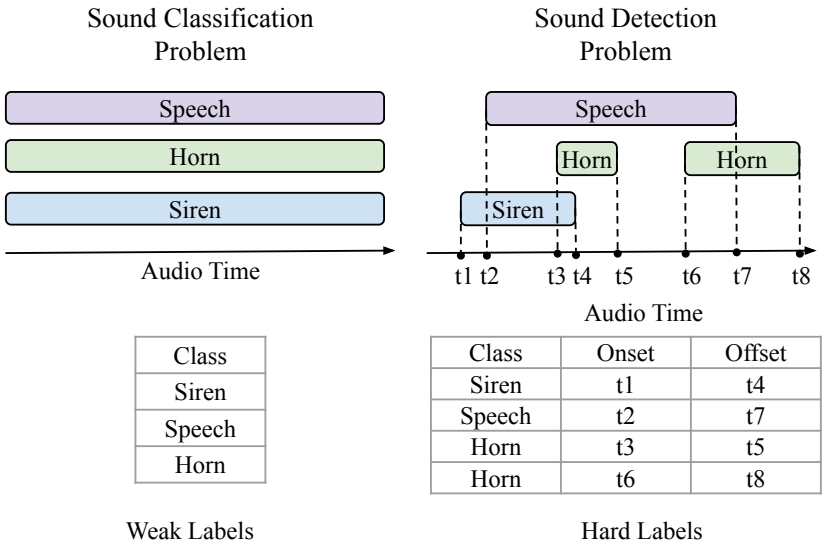


Figure 1: Comparison between sound classification and sound event detection.

within an audio segment but do not provide information about its timing. On the other hand, SED requires hard labels, which provide both the event type and its exact onset and offset times [28]. Figure 1 visually contrasts sound classification and sound event detection, highlighting the difference between weak and hard labels. On the left side, the figure depicts a classification scenario where three sound events span the entire audio segment. The weak labels indicate the presence of these events without providing any temporal details. In contrast, the right side illustrates a detection scenario, where the same three events are precisely segmented with their onset and offset times. This distinction captures the essence of hard labels: they not only identify the event classes but also specify when each event starts and ends.

An interesting aspect of labeling is the ability to convert between hard and weak labels. Since hard labels inherently contain weak label information—indicating whether a sound is present in a segment—it is possible to derive weak labels from hard-labeled data by ignoring the temporal annotations [32]. Conversely, weak labels can sometimes be converted into approximate hard labels when the audio is segmented into sufficiently small time windows. By reducing

the segment duration or using sliding windows with high temporal resolution, it becomes possible to approximate the onset and offset of sound events [33, 34].

Despite their differences, SEC and SED share many commonalities, particularly in their use of machine learning and deep learning techniques [35, 36]. Both tasks rely on transforming raw audio signals into structured representations, such as spectrograms, which serve as input features for deep learning models. Convolutional neural networks (CNNs) are widely used in both SEC and SED due to their effectiveness in capturing local spectral-temporal patterns, while hybrid architectures, such as convolutional recurrent neural networks (CRNNs), combining CNNs with recurrent neural networks (RNNs) or attention mechanisms, are often employed to enhance temporal modeling. Evaluation is another common point between SEC and SED. The metrics used to assess performance in both tasks show similarities due to their shared progressive nature. In fact, many of the evaluation methodologies developed for SEC serve as a foundation for SED.

2.2 General Deep Learning Audio Modeling Process

Building upon the previous discussion of sound event tasks, this section introduces the general process of deep learning-based audio modeling, which generally consists of a sequence of interconnected stages, designed to progressively transform raw audio data into meaningful predictions. While variations exist depending on the task and model architecture, this structure is widely used across numerous works in the field. From environmental sounds [37] to medical applications [38], emotion recognition [17] and industrial noise [39], this pipeline can be adapted to various audio tasks, including classification, detection, and more advanced applications.

The process typically begins with preprocessing, where raw audio signals are cleaned and standardized to ensure consistency. The audio is then transformed into a time-frequency representation, such as spectrograms or Mel spectrograms,

which make it easier to extract relevant patterns [40].

Next, the core of the deep learning model is divided into two main components: feature extraction and classification or detection. The feature extraction block learns to capture essential characteristics — like pitch, texture, and temporal patterns — directly from the input representation. This block often relies on convolutional layers to detect local patterns or recurrent/attention layers for sequential context. The subsequent classification/detection block specializes in interpreting these features to produce the desired output: a class prediction or temporal event detection, depending on the task. In many architectures, these two blocks are part of the same model and are trained jointly, allowing the feature extraction layers to adapt to the requirements of the downstream task.

At this point, the model can be formally defined, for a single input $\mathbf{I}_i \in \mathbb{R}^n$, as:

$$\hat{\mathbf{y}}_i = \mathcal{K}(\mathbf{I}_i; \omega) \tag{1}$$

where $\hat{\mathbf{y}}_i \in \mathbb{R}^d$ represents the model's prediction, $\mathcal{K}(\cdot)$ is a parameterized function governed by the learnable variables ω, and $\mathbf{I}_i$ is the input for the i-th sample.

During training, the model learns to minimize a predefined loss function $\mathcal{L}(\cdot)$ by adjusting its parameters ω via backpropagation and gradient descent. Instead of processing the entire dataset at once, the model is typically trained using mini-batches — small subsets of the data — which help stabilize learning and improve computational efficiency.

The final evaluation stage assesses the performance of the model on unseen data. For both classification and detection tasks, segment-based metrics — such as accuracy or F1-score — provide an overall measure of how well the model identifies sound events within predefined intervals. In detection tasks, event-based metrics are also essential. These metrics evaluate the model's ability to accurately capture individual event instances, considering both the class and the timing precision (onset and offset) [41].

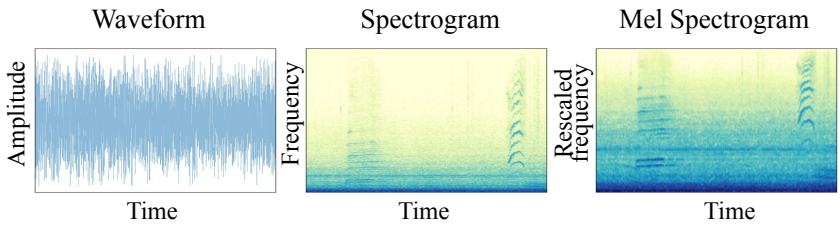


Figure 2: Visual representation of the raw sound waveform, its spectrogram, and Mel spectrogram.

2.2.1 Audio Representation

As highlighted in the previous sections, both SEC and SED tasks require transforming raw audio signals into structured representations that can be effectively processed by deep learning models. Raw waveforms, while rich in information, are difficult to interpret directly due to their high dimensionality, variability, and the continuous nature of the data. To address these challenges, audio signals are typically transformed into time-frequency representations, which provide a more manageable and informative input for machine learning models. This transformation is a common step in both SEC [40, 42, 35, 43] and SED [44, 43, 45] tasks, allowing models to better capture both the temporal and spectral characteristics of the sound.

The two basic representations — spectrogram, and Mel spectrogram — each capture distinct characteristics of the audio signal, making them more or less suitable for different tasks. These representations, as shown in Figure 2, offer complementary views of the audio signal, each emphasizing features suited for particular modeling tasks. The following sections will delve deeper into their construction and specific applications.

Spectrogram

One of the most widely used time-frequency representations is the spectrogram, which illustrates how the frequency content of an audio signal evolves over time. The spectrogram is computed by applying the Short-Time Fourier Transform

(STFT) [46, 47] to the audio signal. Considering a discrete-time signal $\mathbf{x} = [x[0], x[1], \ldots, x[L-1]]$, where $x[n]$ denotes the n-th sample and L is the total number of samples, the STFT segments the signal into short overlapping frames of length N_w. For each frame, a Discrete Fourier Transform (DFT) is computed to obtain its frequency coefficients. The DFT of the m-th windowed frame is given by:

$$X_{m,k} = \sum_{n=0}^{N_w-1} x[n+mH] \cdot w[n] \cdot e^{-j\frac{2\pi}{N_w}kn}, \quad \text{for } k = 0, \ldots, N_w - 1. \tag{2}$$

Here, $X_{m,k} \in \mathbb{C}$ represents the complex STFT coefficient for the m-th frame and k-th frequency bin, N_w is the window length, H is the hop size (the number of samples between the start of adjacent frames), and $w[n]$ is the window function (typically Hanning or Hamming) that reduces spectral leakage. The total number of resulting time frames T is given by $T = \frac{L-N_w}{H} + 1$.

To obtain a finer sampling of the frequency spectrum, the windowed signal frame is often zero-padded to a length N_F (where $N_F \geq N_w$) before computing the DFT. This zero-padding results in N_F frequency bins for each frame, which allows for a more detailed visualization of the spectrum. The result is a matrix $\mathbf{X} \in \mathbb{C}^{T \times N_F}$ with time on one axis and frequency on the other, capturing both temporal and spectral features essential for understanding sound events.

The *power spectrogram* is then typically derived from the STFT by computing the squared magnitude of its complex coefficients:

$$S_{m,k} = |X_{m,k}|^2. \tag{3}$$

This operation transforms the complex-valued STFT coefficients into a real-valued, non-negative time-frequency representation. The resulting spectrogram, $\mathbf{S} \in \mathbb{R}_+^{T \times N_F}$, captures both short-term spectral content and the evolution of sound across time, which is particularly useful for identifying distinct audio events in SEC and SED tasks. The spectrogram has been widely employed in various audio tasks, particularly in early sound classification works such as [40, 42], where it was used to distinguish between different environmental sound categories. Furthermore, the ability of the spectrogram to represent complex frequency patterns over time makes it suitable for tasks such as audio scene

analysis and sound event detection [44].

Mel Spectrogram

While the standard spectrogram uses a linear frequency scale, which is well-suited for many applications, it does not always align with human auditory perception. Human hearing tends to be more sensitive to lower frequencies than higher ones, and this mismatch can be problematic for tasks such as speech recognition or sound classification. To address this, the Mel spectrogram is often used, as it spaces frequency bins according to the Mel scale, which better reflects how the human ear perceives pitch [48]. The Mel scale is defined as:

$$f_{\text{mel}}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{4}$$

where f is linear frequency in Hz, and $f_{\text{mel}}(f)$ is the corresponding Mel frequency.

To generate the Mel spectrogram, the power spectrogram $\mathbf{S}$ is passed through a set of triangular filters uniformly spaced according to the Mel scale. The Mel spectrogram coefficients are then calculated for each frame m as:

$$M_{m,i} = \sum_{k=0}^{N_F-1} G_i(k) S_{m,k}, \tag{5}$$

where $M_{m,i}$ represents the Mel spectrogram coefficient for the m-th time frame and i-th Mel frequency bin. $G_i(k)$ denotes the weight of the i-th triangular Mel filter at linear frequency bin k, and N_F is the total number of linear frequency bins in the power spectrogram $\mathbf{S}$. The filters $G_i(k)$ are designed to have a triangular shape, with their center frequencies and bandwidths aligned with the Mel scale. The resulting Mel spectrogram $\mathbf{M}$ is a real matrix of size $T \times B_{\text{mel}}$, where T is the total number of time frames and B_{mel} is the total number of Mel frequency bins, providing a perceptually relevant representation of the audio signal.

Compared to the spectrogram, the Mel spectrogram compresses the frequency resolution in a way that mimics human hearing [48], allowing models to focus on the most critical frequency bands for a given sound event. This makes it particularly useful for tasks like speech recognition [19] or sound classification in

environments with noise [49], as it emphasizes perceptually significant features while discarding less relevant information.

To further compress the dynamic range of the Mel spectrogram and better reflect the nonlinear sensitivity of human loudness perception, a logarithmic transformation is often applied. The resulting representation is known as the *log-Mel spectrogram* and is computed by applying a logarithm to each element of the Mel spectrogram:

$$\widetilde{M}_{m,i} = \log(M_{m,i} + \epsilon), \tag{6}$$

where ϵ is a small constant added to avoid numerical instability when $M_{m,i} \approx 0$. The log-Mel spectrogram preserves the perceptually meaningful frequency structure of the Mel scale while enhancing lower-energy components, making it a standard input representation for deep learning models in audio-based classification and detection tasks.

2.2.2 Feature Extraction Block

The feature extraction block is a crucial stage in deep learning models for sound classification and detection, transforming input audio representations —such as spectrograms or Mel spectrograms— into a more compact and informative set of features that facilitate subsequent predictions. Depending on the chosen approach, this feature extraction can be performed using various architectures, each leveraging its own strengths. CNNs excel at capturing local spatial patterns in the input data [12, 50], while RNNs, including variants like gated recurrent units (GRUs), are designed to model the temporal evolution of the signal [51, 52]. Meanwhile, transformer-based models have emerged as a powerful alternative, employing self-attention mechanisms to capture both local and global audio structures simultaneously [53, 54, 42]. The choice of a specific architecture —or a combination of them— depends on the model's objectives and the nature of the data, enabling the construction of systems that prioritize spatial representation, temporal patterns, or a balanced integration of both. This flexibility has driven the development of innovative models capable of adapting to a wide range of audio recognition tasks, from acoustic scene classification to precise sound event

detection.

Local Features

CNNs are one of the most effective architectures for extracting local features in sequential data, such as audio signals. These networks operate in both temporal and spectral domains by applying filters (or kernels) that allow the identification of local patterns within the input signal, such as subtle changes in frequency, pitch, or sound texture. These patterns are crucial for tasks such as classification and detection of specific sounds [12, 50, 55].

In the context of audio processing, a two-dimensional (2D) convolutional layer is commonly applied to spectral representations such as the Mel spectrogram. In this process, a kernel—a small matrix of learnable parameters—slides over the input and performs a convolution operation: it multiplies its values by the corresponding values in the section of the spectrogram it covers, sums the result, and adds an additional term called bias, which is also learned during training [56, 57]. This operation is repeated across the entire input with a step size determined by the stride, which defines how far the kernel moves at each iteration. Common stride values include 1 (full overlap) or larger values when dimensionality reduction is desired.

A common technique in this process is padding, which involves adding zeros to the borders of the input to control the output dimensions. This is especially useful for preserving the original dimensions of the spectrogram after convolution. For instance, using a 3×3 kernel without padding would reduce the dimensions by two units in each direction; by applying appropriate padding, the output size can be preserved or adjusted as needed.

Figure 3 illustrates this process, showing a section of a Mel spectrogram to which a 3×3 kernel is applied with a stride of 1 and padding to maintain the input dimensions. The figure highlights which specific input values are involved in the convolution operation that produces each output value. Furthermore, it emphasizes that the same kernel (i.e., set of weights) is reused across all positions in the spectrogram, allowing the detection of similar local patterns regardless of

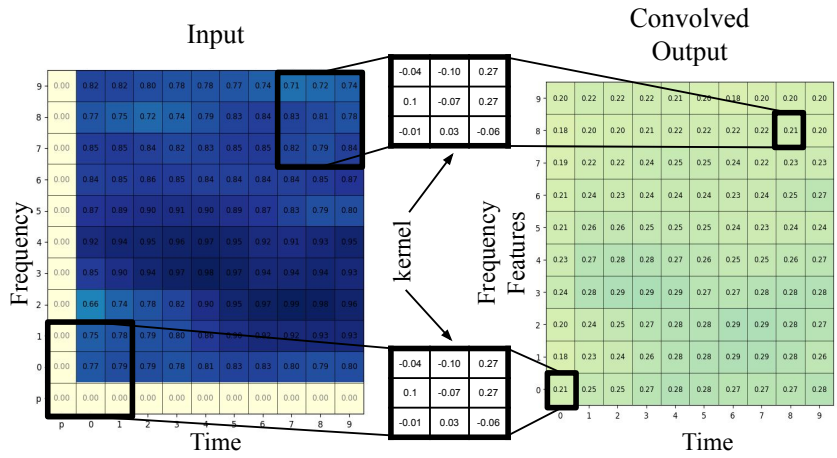


Figure 3: Example of a 2D convolution applied to a Mel spectrogram using a 3×3 kernel with stride 1 and padding.

their spatial location.

Following the convolution operation, it is common to apply a nonlinear activation function, which enables the model to capture more complex relationships in the data. One of the most widely used is ReLU (Rectified Linear Unit), which replaces negative values with zero, introducing nonlinearity and helping to mitigate gradient saturation. Another option is the GLU (Gated Linear Unit), which splits the output of the convolution into two halves: one is modulated by a sigmoid function that acts as a gate over the other, enabling more precise control over the flow of information. The choice of activation function can significantly affect model performance depending on the task and the architecture employed.

After the convolution and activation operations, the resulting output is known as a feature map, which contains the representations generated by the kernel as it scans across the entire input. This representation often retains high-resolution information, much of which may be redundant or unnecessary for the subsequent layers of the model.

To reduce this redundancy and control the dimensionality, an operation known as pooling is applied. Pooling involves dividing the feature map into small,

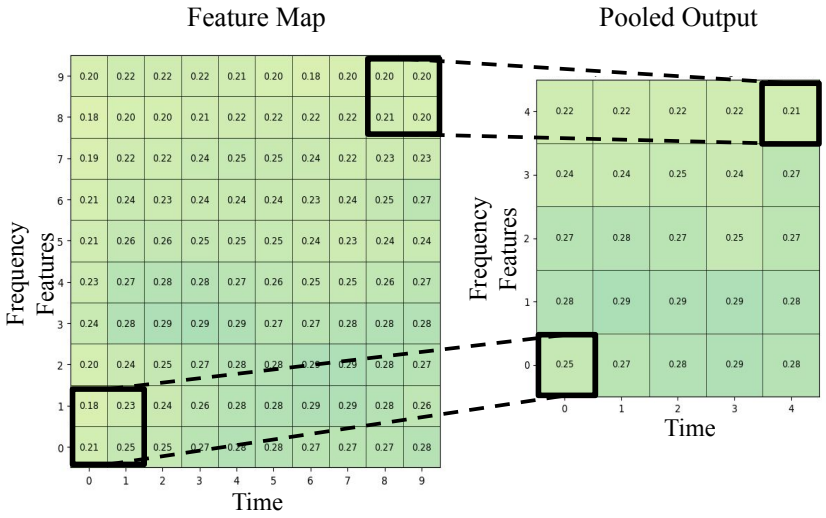


Figure 4: Max pooling operation applied to a feature map with 2×2 non-overlapping regions.

non-overlapping regions (e.g., 2×2) and performing a reduction operation over each region [58, 59]. The most common variant is max pooling, which selects the maximum value within each block, retaining only the most prominent activation. Another variant is average pooling, which computes the average of the values in each region, resulting in a smoother representation.

The main effect of pooling is the reduction of the spatial resolution of the feature map, leading to lower computational cost and introducing a degree of local invariance to small translations or distortions in the input. However, this reduction also implies a partial loss of detailed information. Figure 4 illustrates the max pooling process applied to a feature map, where 2×2 blocks are used and only the maximum value from each is retained, thereby reducing the overall representation size by half along both the temporal and frequency feature axes.

To capture a richer set of local patterns, convolutional layers typically apply multiple filters (kernels) simultaneously. Each filter independently scans the input, generating its own feature map, highlighting specific local activations. These feature maps are stacked together to form a three-dimensional output with dimensions corresponding to time, frequency features, and number of filters.

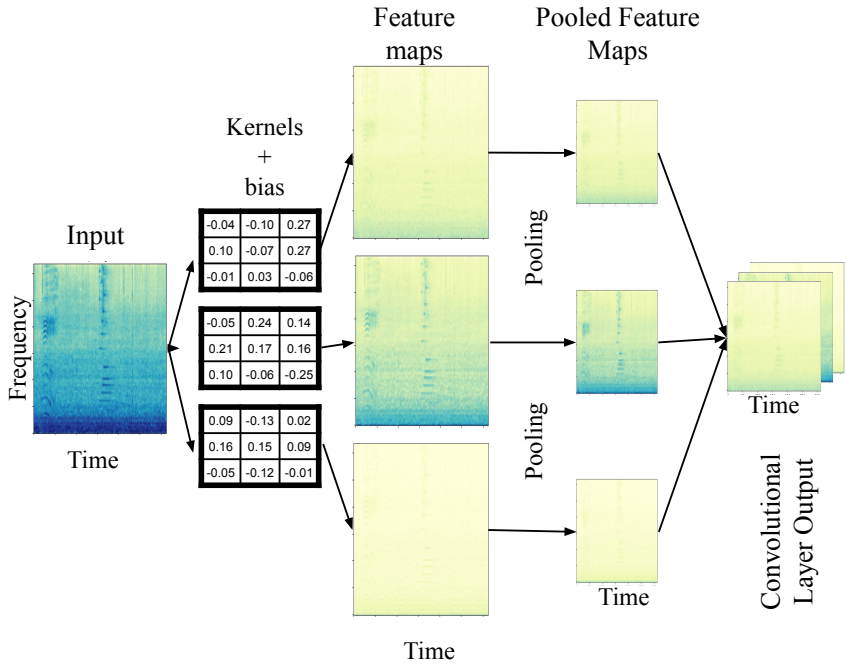


Figure 5: Example of a Mel spectrogram processed by three different convolutional filters, producing three distinct feature maps.

Figure 5 illustrates this process by showing a Mel spectrogram convolved with three distinct filters, producing three separate feature maps. The resulting output captures a diverse set of local patterns across the input.

To improve the training stability and convergence speed of convolutional neural networks, it is common to include a batch normalization layer following the convolution and activation operations. Batch normalization standardizes the feature maps by normalizing their activations across the mini-batch, reducing internal covariate shift and allowing the use of higher learning rates. This technique often leads to better generalization and faster training [60].

Convolutional layers are typically stacked in multiple sequential blocks, where the output feature maps of one layer serve as the input feature maps to the next. As the network deepens, pooling operations progressively reduce the spatial dimensionality (time and frequency resolution) of the feature maps, while the

number of feature maps (filters) usually increases to capture a richer variety of local patterns. This hierarchical structure enables the network to learn increasingly complex and abstract representations. Additionally, the receptive field expands with depth, allowing the integration of information over larger temporal and frequency contexts.

Temporal Features

RNNs and their variants, such as long short-term memory networks (LSTMs) [61] and GRUs [62], are well-suited for modeling temporal dependencies in sequential data, such as audio signals [13, 63, 64]. These architectures operate on sequences in which each timestep corresponds to a feature vector, allowing them to process inputs such as spectrograms, Mel spectrograms, or feature maps extracted from convolutional layers.

Among the most widely used recurrent architectures in audio processing tasks are GRUs, primarily due to their balance between performance and computational efficiency. Compared to LSTMs, GRUs achieve similar modeling capabilities while using fewer parameters, which makes them well-suited for real-time or resource-constrained applications. GRUs rely on gating mechanisms to control the flow of information, enabling them to retain relevant past information and forget irrelevant details over time.

Figure 6 illustrates the operation of a GRU layer, where a sequence of input feature vectors x_t at each time step t is passed through GRU cells. Each cell maintains a hidden state h_t that captures relevant temporal context and is passed along the sequence, allowing the model to incorporate information from previous timesteps when processing future inputs. This temporal recurrence is key for capturing dynamics in audio signals such as rhythm, pitch progression, or phoneme transitions.

To better understand how a GRU processes temporal information, Figure 7 illustrates the internal structure of a GRU cell. At each time step, the cell receives two inputs: the current feature vector x_t and the hidden state from the previous time step h_{t-1}. These elements interact through two main gating mechanisms:

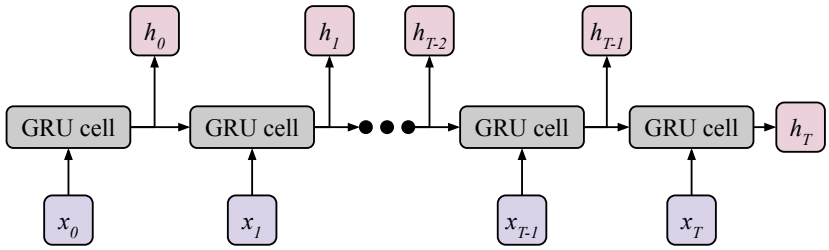


Figure 6: Structure of a GRU layer processing a sequence of input feature vectors.

the update gate z and the reset gate r.

The update gate z controls how much of the past information h_{t-1} should be preserved and passed along to the next state. This gate is computed based on a linear transformation of the current input x_t, with weights W_z, and the past hidden state h_{t-1}, with weights U_z. This mechanism enables the model to maintain long-term dependencies, which is particularly important in audio processing tasks, where relevant patterns may span across long time intervals.

The reset gate r, on the other hand, determines how much of the past state should be forgotten before computing the candidate activation. Like the update gate, it relies on learned weights W_r for the current input and U_r for the previous hidden state. By selectively resetting parts of the memory, this gate allows the GRU to discard irrelevant historical information and focus on the most pertinent aspects of the current input.

Using these gates, the GRU computes a candidate hidden state h'_t, which is a temporary representation combining the new input with the gated past through a nonlinear transformation involving input weights W and recurrent weights U. Finally, the hidden state h_t is updated through an interpolation between the old state and the candidate, guided by the update gate z, providing a flexible memory mechanism.

To enhance the model's ability to capture contextual information from both past and future frames, bidirectional GRUs (Bi-GRUs) are commonly used in audio processing tasks. Unlike standard GRUs that process sequences in a single

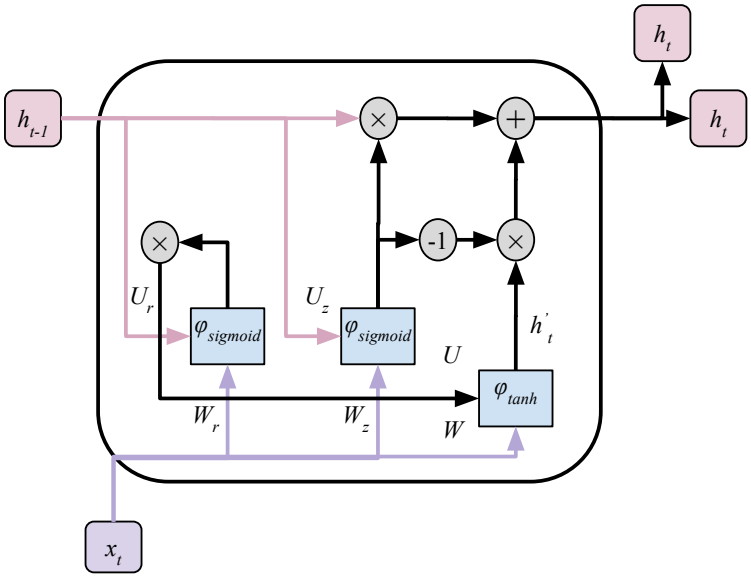


Figure 7: Internal structure of a GRU cell.

forward direction, Bi-GRUs consist of two parallel GRU layers: one processes the input sequence from the beginning to the end (forward pass), while the other processes it in reverse (backward pass). Both layers share the same input at each timestep but maintain separate learnable parameters.

Figure 8 illustrates this architecture using a short segment of a spectrogram. Each timestep, represented as a feature vector extracted from the spectrogram, is fed into two GRU cells running in opposite directions. For the forward direction, the hidden state h_1 is computed based on x_1 and the previous state h_0; in contrast, the backward direction starts from the end of the sequence, where h_{T-1} depends on x_{T-1} and h_T. At each timestep, the forward and backward hidden states are concatenated to form a comprehensive context-aware representation. The dimensionality of each hidden state is determined in part by the size of the linear transformations applied within the GRU cells.

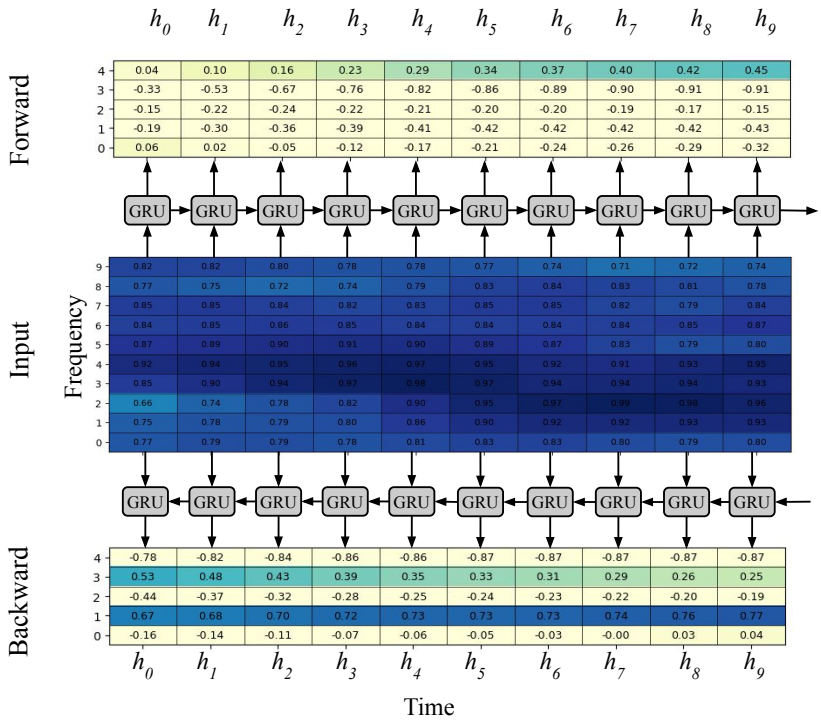


Figure 8: Bidirectional GRU (Bi-GRU) architecture applied to a spectrogram segment.

Transformer-based Features

Transformer architectures have recently emerged as a strong alternative to traditional convolutional and recurrent neural networks for sequential data modeling, including applications in audio signal processing. Originally introduced in the context of natural language processing, Transformers differ fundamentally from CNNs and RNNs in the way they process and represent information. While CNNs focus on detecting local patterns via shared filters, and RNNs sequentially model temporal dependencies through a hidden state passed from one time step to the next, Transformers rely entirely on self-attention mechanisms to dynamically relate all elements of an input sequence to each other [65].

The key innovation in Transformers is the self-attention mechanism, which allows the model to compute dependencies between any two positions in the

input sequence, regardless of their distance. This enables the extraction of both short- and long-range relationships across time and frequency dimensions, without the inductive biases or limitations of locality and sequential processing inherent in CNNs and RNNs. In the context of audio, this means that a Transformer can, in principle, attend to temporally distant yet semantically related acoustic events simultaneously, or relate frequency components that contribute jointly to a perceptual feature, such as timbre or rhythm.

A typical Transformer processes input sequences through several key stages. First, since the architecture itself does not encode any notion of order, positional information is explicitly added to the input embeddings. These embeddings represent each time step (or spectral frame) as a high-dimensional vector, and the positional encodings allow the model to capture the relative or absolute positions of elements in the sequence—an essential feature when working with time-dependent signals like audio.

Once the inputs are embedded and enriched with position information, they are passed through a series of multi-head self-attention layers. In each layer, the model computes attention scores that determine how strongly each element of the sequence should attend to every other element. This operation allows the model to build contextualized representations for each timestep based on the entire sequence, capturing patterns and dependencies that may be far apart in time or frequency.

Each self-attention layer is typically followed by a feed-forward neural network, which refines the representation by applying learned transformations independently to each timestep. To ensure stable training and effective signal propagation, residual connections and layer normalization are applied after both the attention and feed-forward sublayers. These components help mitigate the risk of vanishing gradients and preserve information across layers.

Multiple such layers are stacked to build a deep architecture capable of learning complex interactions in the input. At the output, depending on the task, the model may apply further processing such as pooling, sequence aggregation, or classification heads. In the context of audio, Transformers can be used to map an entire spectrogram or sequence of features into task-specific representations for

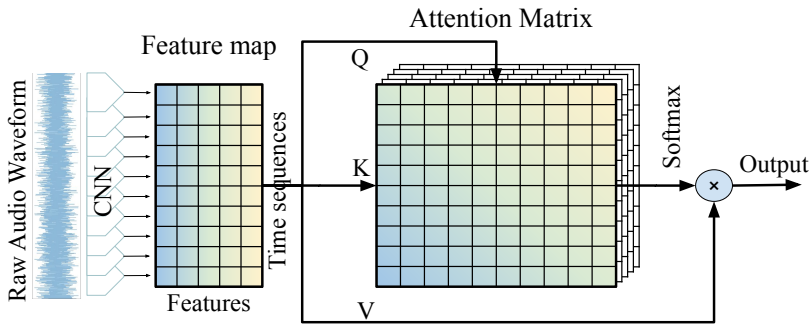


Figure 9: Overview of the wav2vec feature extraction and self-attention mechanism.

speech recognition, sound classification, or acoustic scene understanding.

In many practical audio processing pipelines, Transformers are often combined with convolutional neural networks to leverage the strengths of both architectures. CNNs are effective at extracting robust local features from raw audio or spectrogram inputs by capturing local time-frequency patterns, while Transformers excel at modeling long-range dependencies and global context across the entire sequence. This hybrid approach allows the model to first reduce dimensionality and highlight salient features through convolutional layers before applying self-attention mechanisms to capture complex temporal and spectral relationships.

A prominent example of this synergy is seen in models like wav2vec 2.0 [14], which integrates convolutional feature extraction with Transformer-based contextual modeling. wav2vec 2.0 processes raw audio waveforms by first applying CNNs to generate latent feature representations, which are then fed into a Transformer network to learn context-aware embeddings. This architecture has demonstrated state-of-the-art performance in various speech recognition tasks, showing how combining CNNs and Transformers can effectively handle the hierarchical and sequential nature of audio data.

Figure 9 illustrates the main extraction and attention process in wav2vec. First, the audio signal passes through several convolutional layers that extract

a feature map. This feature map is represented over time and a set of features. From this feature map, three key matrices are generated: K (keys), Q (queries), and V (values). The Q and K matrices are used to compute an attention matrix, which measures the similarity between different temporal positions in the audio. This matrix is passed through a softmax function to convert the values into probabilities that sum to 1, highlighting the most relevant positions. Finally, the soft attention matrix is multiplied by V, the values matrix, to obtain a weighted combination of relevant features at each time step.

2.2.3 Classification and Detection Block

After extracting local, temporal, or transformer-based representations from the input signal through CNNs, RNNs, and transformers, the final stage of the architecture focuses on interpreting these representations for the target task—be it SEC or SED. This block can be divided into two key components: the classification block and the detection block.

In both tasks, the most common approach involves processing the learned representations through one or more fully connected (dense) layers. This requires that the feature maps be structured appropriately, typically in a time–features format when working with audio signals. If the output comes from a 2D convolutional layer—structured as time, features, and channels (or filters)—the output must be reformatted by flattening or stacking the feature and channel dimensions, yielding a time–features structure compatible with subsequent layers.

In contrast, RNN layers inherently produce output in the time–features format, so no additional reshaping is usually required. When combining local (e.g., CNN) and temporal (e.g., Bi-GRU) features, a similar reshaping procedure must be applied to ensure compatibility with the Bi-GRU input format.

Up to this point, the processing pipelines for SEC and SED are largely similar. However, the architectural design diverges afterward depending on the specific task. From this stage on, the network is conditioned to perform either classification or detection, depending on the objective.

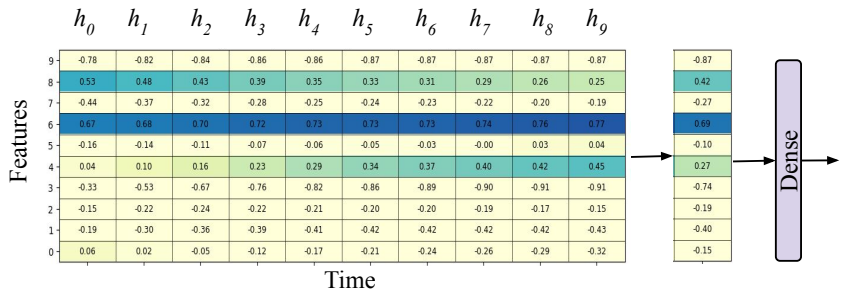


Figure 10: Global Average Pooling applied over time on the output of a Bi-GRU.

Classification Block

For classification tasks, there are several common strategies for handling the learned representations. One of the most widely used approaches is to apply Global Average Pooling across the temporal dimension. This operation averages all values over time, effectively summarizing the entire sequence into a single feature vector. This technique can be applied regardless of whether the final representation comes from a CNN, GRU, or Transformer. Figure 10 illustrates an example using the output of a Bi-GRU, where the sequence of time steps is compressed into a single vector via average pooling.

Another common alternative, particularly when using RNNs such as GRUs, is to use the last hidden state h of the sequence. This state is assumed to encapsulate the contextual information accumulated throughout the input sequence.

Finally, another approach involves applying a flattening operation to the feature map, stacking all time steps into a single long vector. This technique preserves all temporal information at the cost of increased dimensionality. All these strategies produce a fixed-size vector that summarizes the entire input sequence and can then be used for categorical or continuous prediction through fully connected layers.

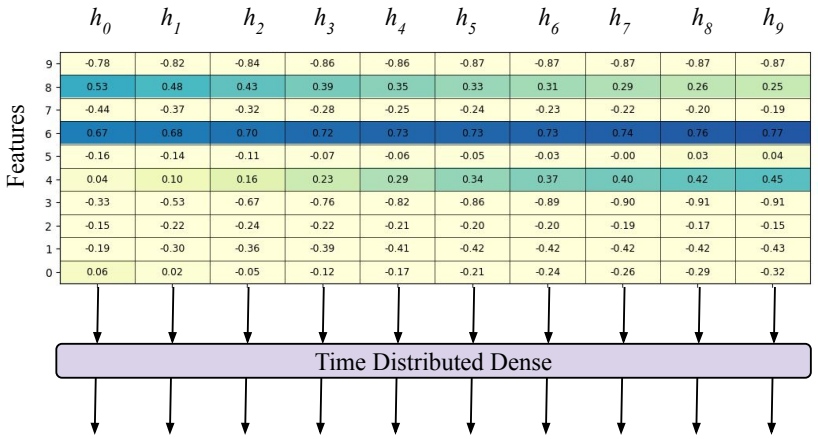


Figure 11: Example of a detection block applied to a feature map generated by a Bi-GRU.

Detection Block

A widely used strategy for event detection in temporal sequences is to leverage the feature maps generated by models such as CNNs, RNNs, or Transformers that preserve the temporal dimension. In this approach, each time step is evaluated independently through a dense layer, allowing a prediction to be made per time step [29, 66].

However, this method involves a significant trade-off between prediction accuracy and computational efficiency. The quality of the predictions is directly influenced by the resolution of the final feature map used. If the resolution is low (i.e., the feature map has fewer time steps), the model becomes more efficient, but the temporal precision of the prediction decreases. On the other hand, using a high-resolution feature map (with more time steps) allows for finer-grained detection, but at the cost of increased computational load due to the larger number of steps that must be processed.

Figure 11 illustrates this process using a feature map generated by a Bi-GRU network as an example. It shows how a dense layer is applied independently at each time step to generate the corresponding event predictions.

Alternatively, in regression-based detection approaches—taking inspiration

from object detection architectures such as YOLO—the model predicts event boundaries directly [45, 16, 9]. Instead of classifying each frame independently, the network outputs global information about event onset, offset, and class as continuous values.

In this case, each time step is also processed independently, similar to the previous strategy. However, rather than outputting a class probability alone, the network predicts both the likelihood that an event occurs at that time step and the corresponding temporal boundaries (i.e., onset and offset times) of the event. This allows for a more compact and direct representation of events, especially useful in scenarios where precise temporal localization is critical.

2.2.4 Model Training

Training is a crucial phase in the development of deep learning models, where the model learns to make accurate predictions by adjusting its internal parameters (ω). This process involves presenting the model with a set of labeled data, allowing it to gradually improve its performance over time. The model iteratively processes batches of data, compares its predictions to the labels, and updates its weights to minimize the error. The goal is to find a set of weights that minimize the discrepancy between the model's predictions and the actual target values.

Mathematically, the learning process can be formulated as the following optimization problem:

$$\omega^* = \arg\min_{\omega} \ \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}(\omega)), \tag{7}$$

where $\mathcal{L}(\cdot)$ is a loss function that quantifies the discrepancy between the ground-truth outputs $\mathbf{Y} \in \mathbb{R}^{N \times d}$ and the model predictions $\hat{\mathbf{Y}}(\omega) \in \mathbb{R}^{N \times d}$ computed over a dataset of N samples with d-dimensional outputs. The predictions depend on the model parameters ω.

This minimization is achieved using an optimization algorithm, most commonly gradient descent and its variants (e.g., stochastic gradient descent, Adam). Gradient descent updates the model's parameters ω by computing the gradient of the loss function with respect to each parameter and taking steps in

the direction that reduces the loss. For a given parameter ω_p, the update rule is defined as:

$$\omega_p \leftarrow \omega_p - \eta \cdot \frac{\partial \mathcal{L}}{\partial \omega_p}, \tag{8}$$

where η is the learning rate, a hyperparameter that controls the step size. The loss $\mathcal{L}$ is typically computed over a mini-batch of training data, which helps stabilize learning and improve computational efficiency. This iterative process continues until convergence, i.e., until the loss reaches a minimum or no longer improves significantly. Gradient descent is central to training deep networks, enabling them to learn complex representations by systematically reducing prediction errors.

In the context of SEC and SED, the selection of an appropriate loss function is essential to guide the training of deep learning models. This section explores three fundamental loss functions: Mean Squared Error (MSE), Binary Cross-Entropy (BCE), and Categorical Cross-Entropy (CCE)—each suited for different types of tasks. Additionally, we discuss two complementary loss functions that target specific challenges in sound classification and detection: Center Loss, which enhances the discriminative power of learned representations by enforcing intra-class compactness and inter-class separation, and the Intersection Loss (CIoU), which improves temporal alignment in event detection by directly penalizing inaccurate onset and offset predictions.

Mean Squared Error (MSE)

The mean squared error (MSE) loss, denoted by $\mathcal{L}_{\text{MSE}}$, is primarily used in regression tasks, but it also arises in classification settings where predictions are treated as continuous values. MSE quantifies the average squared difference between the model's predictions and the ground-truth targets.

Given a single input $\mathbf{I}_i$, let $\mathbf{y}_i \in \mathbb{R}^d$ be its ground-truth target vector, and $\hat{\mathbf{y}}_i \in \mathbb{R}^d$ the corresponding model prediction. The MSE loss for this individual sample is defined as:

$$\ell_{\text{MSE}}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \frac{1}{d}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 = \frac{1}{d}\sum_{j=1}^{d}(y_{ij} - \hat{y}_{ij})^2. \tag{9}$$

During training, the MSE is typically computed over a mini-batch of N samples as the average of the per-sample losses:

$$\mathcal{L}_{\text{MSE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{MSE}}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \frac{1}{N \cdot d} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2, \tag{10}$$

where $\mathbf{Y}, \hat{\mathbf{Y}} \in \mathbb{R}^{N \times d}$ are the matrices containing all ground-truth and predicted output vectors in the batch.

The MSE penalizes large errors more heavily than small ones due to the squared term, making it particularly sensitive to outliers. Although not specifically designed for classification, MSE can still be applied in scenarios involving continuous-valued targets—for example, estimating the intensity or temporal boundaries of sound events in regression-based detection tasks. However, for probabilistic outputs or discrete class predictions, alternative loss functions such as cross-entropy are generally more suitable.

Binary Cross-Entropy (BCE)

The binary cross-entropy (BCE) loss, denoted by $\mathcal{L}_{\text{BCE}}$, is widely used in binary and multi-label classification tasks, where each output unit represents an independent binary decision. In this setting, the model predicts a probability $\hat{y}_{ij} \in [0, 1]$ for the j-th output of the i-th sample, and the corresponding ground-truth label is $y_{ij} \in \{0, 1\}$.

The BCE loss for a single sample $\mathbf{I}_i$ is defined as:

$$\ell_{\text{BCE}}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\frac{1}{d} \sum_{j=1}^{d} \left[y_{ij} \cdot \log(\hat{y}_{ij}) + (1 - y_{ij}) \cdot \log(1 - \hat{y}_{ij}) \right], \tag{11}$$

where $\mathbf{y}_i, \hat{\mathbf{y}}_i \in [0, 1]^d$ are the binary label vector and the predicted probabilities for the i-th sample.

During training, the loss is computed over a mini-batch of N samples by averaging the per-sample losses:

$$\mathcal{L}_{\text{BCE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{BCE}}(\mathbf{y}_i, \hat{\mathbf{y}}_i), \tag{12}$$

where $\mathbf{Y}, \hat{\mathbf{Y}} \in [0, 1]^{N \times d}$ are the label and prediction matrices for the batch.

To ensure numerical stability when predicted probabilities are near 0 or 1, a small constant $\epsilon > 0$ is added:

$$\mathcal{L}_{\text{BCE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\frac{1}{N \cdot d} \sum_{i=1}^{N} \sum_{j=1}^{d} \left[y_{ij} \cdot \log(\hat{y}_{ij} + \epsilon) + (1 - y_{ij}) \cdot \log(1 - \hat{y}_{ij} + \epsilon) \right]. \tag{13}$$

This formulation is suitable for binary classification as well as multi-label scenarios, where each output dimension is treated as an independent binary decision.

Categorical Cross-Entropy (CCE)

For multi-class classification tasks, where each sample belongs exclusively to one of C possible classes, the categorical cross-entropy (CCE) loss, denoted by $\mathcal{L}_{\text{CCE}}$, is commonly used. It generalizes the binary cross-entropy to multi-class settings by comparing the model's predicted probability distribution with the ground-truth class label, represented as a one-hot encoded vector.

Given a mini-batch of N samples, the CCE loss is defined as:

$$\mathcal{L}_{\text{CCE}}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}), \tag{14}$$

where $\mathbf{Y}, \hat{\mathbf{Y}} \in [0,1]^{N \times C}$ are the matrices of ground-truth labels and predicted class probabilities for the batch. Each row $\mathbf{y}_i \in \{0,1\}^C$ is a one-hot vector indicating the correct class for the i-th sample, and $\hat{\mathbf{y}}_i \in [0,1]^C$ is the corresponding probability distribution predicted by the model (typically obtained via a softmax layer).

CCE encourages the model to assign high probability to the correct class while minimizing the probabilities of incorrect ones. This loss function is widely used in sound event classification tasks, where the objective is to assign a single label to each sound event from a predefined set of mutually exclusive categories.

Center Loss

The center loss [67] is designed to enhance the discriminative power of a model by minimizing intra-class variance while maintaining inter-class separability. It

operates on intermediate feature representations by pulling samples of the same class closer to a learned class-specific centroid in the embedding space.

Let $\hat{\mathbf{x}}_i \in \mathbb{R}^d$ denote the feature representation predicted by the model for the i-th sample, and let $\boldsymbol{\mu}_c \in \mathbb{R}^d$ be the centroid of class c. The ground-truth label is assumed to be encoded as a one-hot vector $\mathbf{y}_i \in \{0,1\}^C$.

The center loss is then defined as:

$$\mathcal{L}_C(\mathbf{Y}, \hat{\mathbf{X}}) = \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \left\| \hat{\mathbf{x}}_i - \boldsymbol{\mu}_c \right\|^2, \tag{15}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{N \times d}$ is the matrix of predicted feature representations for a mini-batch, and $\mathbf{Y} \in \{0,1\}^{N \times C}$ is the one-hot encoded label matrix. Each centroid $\boldsymbol{\mu}_c \in \mathbb{R}^d$ represents the mean feature vector of class c in the embedding space. These centroids are learned during training and updated to minimize the intra-class distances:

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i:\, y_i = c} \hat{\mathbf{x}}_i,$$

where N_c is the number of samples in the batch with ground-truth class c.

As center loss alone does not enforce sufficient inter-class separation, it is commonly combined with a classification loss such as categorical cross-entropy. The combined objective encourages both intra-class compactness and inter-class discriminability:

$$\mathcal{L} = \mathcal{L}_C(\mathbf{Y}, \hat{\mathbf{X}}) + \mathcal{L}_{\text{CCE}}(\mathbf{Y}, \hat{\mathbf{Y}}), \tag{16}$$

where $\hat{\mathbf{Y}}$ denotes the matrix of predicted class probabilities.

Intersection Over Union Loss (CIoU)

In sound event detection tasks that involve temporal localization, the Intersection Loss—often referred to as CIoU (Center-based Intersection over Union Loss)—is used to quantify the temporal overlap between predicted and truth label events.

Assuming that event activities are represented by their center (c) and width (w), the segment boundaries for predictions ($p_{\text{start}}, p_{\text{end}}$) and ground-truths ($g_{\text{start}}, g_{\text{end}}$) are defined as:

$$\begin{aligned} p_{\text{start}} &= p_c - \frac{p_w}{2}, \qquad p_{\text{end}} = p_c + \frac{p_w}{2} \\ g_{\text{start}} &= g_c - \frac{g_w}{2}, \qquad g_{\text{end}} = g_c + \frac{g_w}{2} \end{aligned} \tag{17}$$

The intersection and union of the predicted and label intervals are computed as:

$$\text{Intersection} = \max(0, \min(p_{\text{end}}, g_{\text{end}}) - \max(p_{\text{start}}, g_{\text{start}})) \tag{18}$$

$$\text{Union} = \max(p_{\text{end}}, g_{\text{end}}) - \min(p_{\text{start}}, g_{\text{start}}) \tag{19}$$

The loss is then defined as the complement of the intersection-over-union (IoU):

$$\mathcal{L}_{\text{CIoU}} = 1 - \frac{\text{Intersection}}{\text{Union}}. \tag{20}$$

This loss encourages the model to output segments that better align with the true temporal boundaries of events, penalizing both early/late predictions and duration mismatches [68].

2.2.5 Model Evaluation

Evaluating the performance of machine learning models is essential to understanding their effectiveness and reliability. In the context of SEC and SED tasks, various evaluation strategies have evolved, ranging from simple frame-based metrics to more complex event-based approaches. This section explores both methods, highlighting their strengths, limitations, and the progression toward more robust evaluation techniques.

Frame-based Metrics

Frame-based metrics evaluate a model's performance by analyzing predictions at a fixed time resolution. In SEC tasks, the entire audio duration is typically treated as a single instance, and performance is assessed using values like true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). These values are then used to derive traditional evaluation metrics such as:

- *Recall* score (R) measures the model's ability to detect all relevant sound events:

$$\mathrm{R} = \frac{TP}{TP + FN}. \tag{21}$$

  It is especially important when missing an event is more detrimental than producing a false alarm.

- *Precision* score (P) measures the proportion of predicted events that are actually correct:

$$\mathrm{P} = \frac{TP}{TP + FP}. \tag{22}$$

  A high precision indicates that the model generates few false positives, which is particularly relevant in applications where false alarms must be minimized.

- The *F1-score* (F1) is the harmonic mean of precision and recall, offering a balanced perspective between detecting events and avoiding false positives. Expressed using TP, FP, and FN:

$$F1 = \frac{2TP}{2TP + FP + FN}. \tag{23}$$

  It is widely used when the dataset is imbalanced or when both false positives and false negatives are critical.

- *Accuracy* (ACC) quantifies the overall proportion of correct predictions:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{24}$$

  Although intuitive, this metric can be misleading in imbalanced scenarios, where a model may appear accurate by predominantly predicting the majority class.

- The *Geometric Mean* (G-mean) quantifies the balance between the true positive rate (TPR) and the true negative rate (TNR), and is particularly useful in evaluating performance on imbalanced datasets. It encourages the model to perform well across both positive and negative classes. :

$$G\text{-}mean = \sqrt{\text{TPR} \cdot \text{TNR}} = \sqrt{\frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP}}. \tag{25}$$

  This is particularly useful in SEC and SED, where class imbalance is common and performance on minority classes is critical.

- The *Area Under the Curve* (AUC) corresponds to the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR), where FPR = 1 − TNR, as the decision threshold varies. It is formally defined as:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR}. \tag{26}$$

  It reflects the model's ability to rank positive instances higher than negative ones, independently of any specific decision threshold (θ).

When the resolution is reduced to smaller segments—dividing the audio into frames—the same metrics can be applied to more dynamic problems like SED. In this case, each frame is independently classified, and system performance is evaluated by comparing the predicted class for each frame against the corresponding label.

To illustrate how the assignment of TP, FP, TN, and FN differs between SEC and SED tasks, Figure 12 compares both approaches. In SEC, where a single window is used to evaluate the entire audio, the assignment is based on the prediction for the entire event duration. In contrast, in SED, where the audio is divided into multiple smaller windows (in this case, 10 segments), each frame is classified independently, which affects how these values are assigned. For the SED case, each class and frame can be assigned to TP, TN, FP, or FN depending on whether the predicted event aligns with the label within the frame. It is important to note that an event will be considered a TP if the label and the prediction are present in the frame, regardless of their proportion.

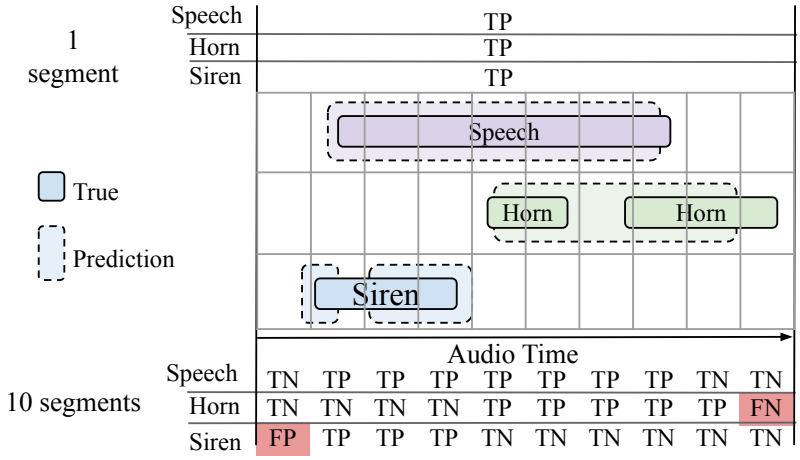


Figure 12: Comparison of TP, FP, TN, and FN assignment in frame-based metrics between SEC and SED tasks.

One advantage of frame-based metrics is their simplicity. They provide detailed insights into how well a model tracks rapid changes in sound classes over time. However, it can also introduce issues. For instance, minor temporal misalignments between predictions and label can lead to significant drops in performance scores, even when the overall event detection is subjectively acceptable.

Moreover, frame-based metrics treat each frame as an isolated decision, ignoring the temporal structure of sound events. This can lead to fragmented detections, where an event is partially recognized but interrupted by brief periods of incorrect classifications, penalizing the model more harshly than might be perceptually reasonable.

Event-based Metrics

The evaluation of SED systems has evolved significantly, moving from early frame-based approaches to more sophisticated metrics aimed at capturing the overall performance of the system. Traditionally, evaluation metrics have relied on the collar method to determine whether a detection is correct. A collar defines a time window around the start and end of a labeled event. If the detection

falls within this window, it is considered a TP. Otherwise, it is classified as a FP or a FN [69, 70]. However, this approach introduces limitations, especially when events are difficult to delimit precisely or when there is disagreement among human annotators. For example, a continuous dog bark could be interpreted as one long event or multiple short events, and using collars would force a single interpretation, penalizing the other.

To overcome these limitations, the polyphonic sound detection score (PSDS) was proposed as a more robust metric against labeling subjectivity and variability in system operating points [41]. PSDS redefines true positives and false positives using more flexible intersection-based criteria. The detection tolerance criterion (DTC) evaluates the amount of intersection between the detection and the ground truth event, ensuring that a significant part of the detected event matches the ground truth. Meanwhile, the ground truth intersection criterion (GTC) requires a fraction of the ground truth event to overlap with the detection to be considered a true positive. This strategy allows partially detected events to be recognized as valid hits instead of being penalized as errors.

To visually illustrate these differences, Figure 13 compares traditional collar-based evaluation and the PSDS approach. On the left, true positives are determined only if the detection falls within a fixed window around the ground truth event, which can penalize valid detections that do not meet the expected precision. On the right, the PSDS approach allows a detection to be considered valid if there is sufficient intersection between the detection and the ground truth event, regardless of the exact boundary alignment. This more flexible method better reflects system performance.

In addition to DTC and GTC, PSDS incorporates the cross-trigger tolerance criterion (CTTC), which aims to identify false positives that coincide with ground truth events of a different class than the one predicted. This is especially useful in multi-class systems where certain classes may be acoustically similar. The CTTC helps separate these errors from conventional false positives, providing a more detailed evaluation and better reflecting potential data biases or class confusions.

Finally, PSDS synthesizes the system's overall performance by calculating the area under the curve (AUC) of a specific curve for polyphonic detection

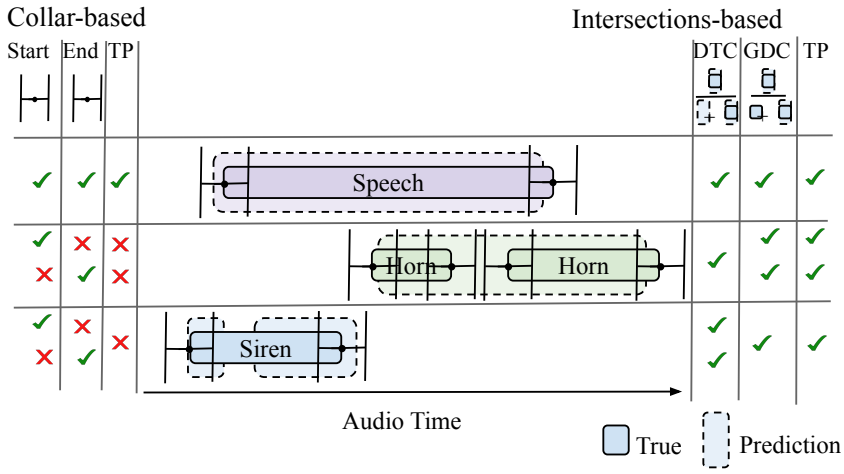


Figure 13: Comparison of TP, FP, TN, and FN assignment in event-based metrics between Collar-based and intersection-based approaches.

(PSD-ROC). This curve summarizes the TPR against the FPR across different operating points, providing a single, representative performance score. In this way, PSDS not only measures how well the system detects events but also how stable its performance remains under various configurations. Moreover, by having the TP, FP, and FN values at the event level, it is possible to derive traditional metrics such as Recall or F1, enabling comparisons with more conventional approaches.

2.3 Common Challenges and Solutions

Both SEC and SED face a variety of challenges, many of which overlap due to the similarities between the tasks. While SEC focuses on classifying sound events over a fixed duration and SED aims to identify and localize these events in time, both tasks encounter obstacles related to data quality, class imbalance, generalization, and resource optimization. The complexity of audio data—affected by environmental noise, overlapping events, and varying acoustic characteristics—exacerbates these challenges. These issues are actively addressed in the current state of the art, with various strategies proposed to mitigate their impact. Understanding these common difficulties and the corresponding solutions

is crucial, as they directly inform the methodologies employed in the development of this thesis and contribute to improving model performance in both areas.

2.3.1 Data Acquisition

Acquiring high-quality labeled data is a fundamental challenge in both SEC and SED tasks. One of the primary obstacles is the difficulty of obtaining recordings from diverse and specific real-world environments, such as urban traffic or driving scenarios. Accessing these types of environments can be challenging due to the rarity or inaccessibility of certain contexts. Furthermore, recording high-quality audio in these environments is a time-consuming and resource-intensive task that requires careful planning, equipment, and often the coordination of fieldwork efforts. The complexity of capturing sound events in these varied settings makes it difficult to obtain large amounts of representative data, which in turn impacts model training.

The quality of the labels also heavily impacts the performance of the model [71, 72]. In many cases, human annotators are required to manually identify the onset and offset of sound events. However, this process is prone to errors, including mislabeling, inconsistent annotation practices, and subjective interpretations [73]. Additionally, for sound events that are subtle or have overlapping characteristics, achieving reliable labels becomes even more challenging, which complicates model training.

To address these challenges, one effective solution is the generation of synthetic data. This process involves combining isolated sound samples with background noise to create realistic audio mixtures. Typically, a base track of ambient noise is selected, and various isolated sound sources are then added—often in a randomized manner—to simulate diverse acoustic environments. By leveraging this approach, researchers can generate large quantities of labeled data without requiring manual annotation, helping to overcome limitations in real-world data availability. Additionally, synthetic data can augment existing datasets, introducing variability and improving the robustness of models to different acoustic conditions. The overall process of generating synthetic data is illustrated

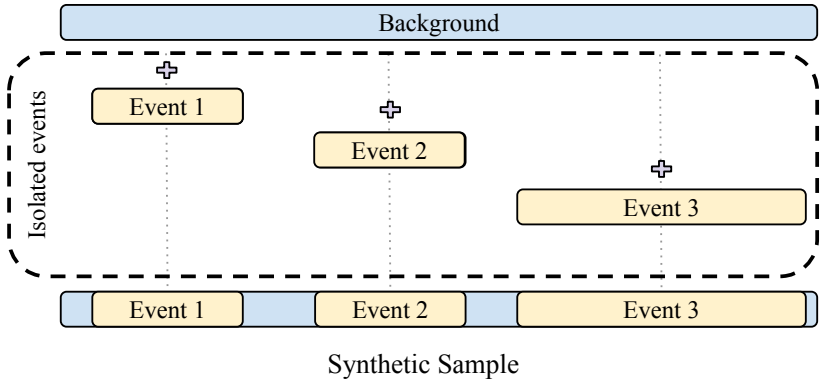


Figure 14: Illustration of the synthetic data generation process.

in Figure 14, where individual sound events are blended with a background noise track to create a diverse set of training samples.

Another promising approach is the use of soft labels [66]. Unlike hard labels (e.g., binary labels for the presence or absence of an event), soft labels reflect the uncertainty in the annotations. For instance, rather than strictly defining the start and end times of an event, soft labels assign probabilities to different time intervals, capturing the ambiguity inherent in many sound events. This enables the model to learn from imperfect or imprecise annotations, improving its generalization and robustness in cases where perfect labels are difficult to obtain.

2.3.2 Class Imbalance

In addition to the challenges inherent in acquiring high-quality data and precise annotations, another significant issue affecting both SEC and SED related to the data is class imbalance. This phenomenon not only complicates the quality of model training but also introduces substantial biases in its performance and evaluation. First, models tend to prioritize the most frequent classes, resulting in poorer representation of less common events. This means the model learns to recognize the dominant classes well, but struggles with minority classes, leading to under-representation in predictions. Second, imbalance distorts performance metrics. For example, a model biased toward majority classes may achieve high

overall accuracy, but this masks poor recall and F1 for the minority classes —giving a misleading impression of performance.

This problem becomes even more pronounced in SED due to its temporal nature. In addition to differences in class appearance frequency, event duration introduces a second layer of imbalance. Short-duration events are inherently less represented —not only in terms of the number of examples but also in the total number of frames they occupy [74]. As a result, the model may overlook these brief events, further lowering recall and contributing to fragmented detections.

One solution to this problem is the application of weighting mechanisms. By assigning higher importance to underrepresented classes or events with shorter durations, the model becomes more sensitive to these cases. This approach can be integrated into the loss function by incorporating class-specific weighting factors, ensuring that misclassifications of minority events are penalized more heavily [75, 74].

To address class imbalance in binary or multi-label classification tasks, the BCE loss defined in Equation (11) can be modified to include a weighting factor γ that adjusts the contribution of positive labels for each output dimension. The weighted BCE loss for a single sample becomes:

$$\ell_{\mathrm{BCE}_\gamma}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\sum_{j=1}^{d} \left[\gamma\, y_{ij} \log(\hat{y}_{ij} + \epsilon) + (1 - y_{ij}) \log(1 - \hat{y}_{ij} + \epsilon)\right]. \tag{27}$$

The batch-level loss is then computed as the average over N samples:

$$\mathcal{L}_{\mathrm{BCE}_\gamma}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N} \sum_{i=1}^{N} \ell_{\mathrm{BCE}_\gamma}(\mathbf{y}_i, \hat{\mathbf{y}}_i). \tag{28}$$

Similarly, for multi-class classification, the categorical cross-entropy (CCE) loss from Equation (14) can be extended with a class-specific weighting factor γ_c:

$$\mathcal{L}_{\mathrm{CCE}_\gamma}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \gamma_c\, y_{i,c} \log(\hat{y}_{i,c}). \tag{29}$$

In these formulations, γ adjusts the relative importance of positive versus negative labels in binary or multi-label settings, while γ_c allows emphasis on specific classes in multi-class classification. This strategy helps improve performance on

underrepresented classes and contributes to balancing metrics such as recall and F1-score.

Another solution is data preparation. Creating balanced datasets, either by oversampling minority classes or downsampling majority ones, can mitigate imbalance. This involves adjusting the dataset to ensure all event types —regardless of frequency or duration— are well-represented. Additionally, generating synthetic data (as discussed in the previous section) can further reinforce underrepresented classes, promoting better model learning and producing more reliable evaluation metrics.

2.3.3 Model Generalization and Overfitting

Generalization is a crucial aspect of machine learning, as it determines how well a model can perform on unseen data. In both SEC and SED tasks, the risk of overfitting—where a model learns the specific details of the training data but fails to generalize to new examples—is a significant concern.

One effective strategy to mitigate overfitting is the use of the Exponential Moving Average (EMA) technique, employed in models like Mean Teacher [76, 77]. The EMA technique helps stabilize the model's predictions over time by smoothing the updates to the parameters. Given a model parameter ω_t at training step t, its EMA update is defined as:

$$\omega_t^{EMA} = \beta\omega_{t-1}^{EMA} + (1-\beta)\omega_t, \tag{30}$$

where β is a decay factor that controls the influence of past values, typically set close to 1. By averaging the parameters over multiple training steps, EMA reduces the sensitivity of the model to fluctuations in the data, leading to improved generalization and more stable predictions.

Additionally, data augmentation techniques are widely used to combat overfitting [78]. By artificially increasing the diversity of the training data, these techniques help improve model robustness and generalization. Common augmentations in audio include transformations such as time stretching and pitch shifting, which are typically applied to isolated sound events to generate synthetic

variants.

Time stretching modifies the speed of an audio signal without affecting its pitch, allowing the model to learn from events with varying durations. Pitch shifting, on the other hand, alters the tonal content of the audio by changing its frequency components, enabling the model to generalize across different pitch ranges.

Unlike these preprocessing techniques, mixup is applied dynamically during training. It creates new training examples by interpolating pairs of inputs and their corresponding labels within each mini-batch. This regularization strategy encourages the model to behave linearly between training samples, which reduces overfitting and enhances generalization.

Another important strategy for improving generalization is *domain adaptation* [79]. Models trained on a specific dataset often struggle to perform well when deployed in different acoustic environments or with different recording devices. Domain adaptation techniques aim to minimize the gap between the source (training) and target (deployment) domains. This can be done through feature alignment, where the model learns to extract domain-invariant representations, or via adversarial training, where a domain discriminator guides the model to learn features that generalize across domains.

2.3.4 Resource Optimization

Resource optimization is essential for deploying SEC and SED models, particularly in real-time applications where low latency and computational efficiency are crucial. Two widely adopted techniques to achieve this are pruning and quantization, both of which aim to reduce the model's size and computational demands without significantly compromising performance.

Pruning involves selectively removing weights, neurons, or even entire layers that contribute minimally to the model's predictions [80, 81]. This results in a smaller, faster model that maintains an acceptable performance level. Static pruning is typically applied after training, creating a leaner, fixed architecture, while dynamic pruning occurs during training, allowing the model to adaptively

refine its structure based on the data and gradients. Dynamic approaches can yield more efficient models by learning which parts of the network can be discarded early on.

Quantization, on the other hand, reduces the precision of the model's numerical parameters. Instead of using 32-bit floating-point numbers, weights and activations are converted to lower-bit representations, such as 8-bit integers [82, 83]. Static quantization applies this transformation after training, leading to a compressed model that remains consistent across all inputs. Dynamic quantization goes further by adjusting precision during inference based on the input data, enabling faster computations while preserving accuracy. Combining quantization with pruning can further amplify the gains, creating lightweight models that are suitable for deployment on resource-constrained devices like embedded systems or edge devices.

A practical way to apply these techniques is through ONNX (Open Neural Network Exchange) [84], an open-source framework designed to enable interoperability between different deep learning environments like PyTorch, TensorFlow, and Caffe. ONNX supports both pruning and quantization techniques, offering optimized runtimes for diverse hardware architectures. By converting trained models into ONNX format, developers can deploy them efficiently across a variety of platforms, ensuring seamless integration and performance gains without extensive re-engineering. This makes ONNX an attractive solution for bringing optimized SEC and SED models from research to real-world applications.

Chapter 3:
A Framework for Acoustic Driving Environments

This chapter presents one of the key contributions of this thesis: a comprehensive framework designed to analyze driving environments from an acoustic perspective, with a focus on SEC and SED applications, particularly those related to safety and driver distractions.

At the core of this framework is a detailed taxonomy that classifies relevant sound events in driving contexts. This taxonomy covers both internal sounds —such as audio systems and passenger conversations— and external signals, including emergency sirens and environmental disturbances. This structured foundation helps interpret the acoustic landscape during driving and supports the identification of key sound events that may impact driver safety.

The framework also addresses the role of acoustic distractions, focusing on sounds that are particularly disruptive to drivers. Key examples include mobile phone notifications, passenger interactions, and emergency vehicle sirens. This analysis is backed by existing research highlighting how such events impair driver attention and increase accident risk.

To support the development and evaluation of advanced SEC and SED systems, the framework incorporates a collection of specialized datasets. Building upon the proposed taxonomy and the identification of key acoustic distractions, these datasets are generated using both synthetic and real audio recordings, simulating diverse driving conditions. This approach not only promotes the development of more robust and practical methodologies for real-world applications but also ensures that the most safety-critical sound events are properly represented. Additionally, a specific dataset explores drivers' emotional states, acknowledging the influence of emotional context on driving behavior and overall safety.

By integrating a structured taxonomy, an analysis of critical acoustic distractions, and carefully designed datasets, this framework provides essential tools for applying SEC and SED techniques to driving environments. It enables the identification and classification of key sound events relevant to driver awareness and safety, facilitating the adaptation of these tasks to the unique acoustic challenges present in driving scenarios.

3.1 Acoustic Driving Environment

Understanding and organizing the complex acoustic landscape of driving environments is crucial for developing effective driver monitoring and sound analysis systems. The diversity of sounds present both inside and outside the vehicle presents a significant challenge, which is why we propose a comprehensive and systematic taxonomy that classifies the wide range of sounds found in and around vehicles. This taxonomy not only supports academic research by providing a standardized framework for categorizing driving-related sounds, but it also serves as a foundation for the design and implementation of SEC and SED systems aimed at improving driver awareness, safety, and comfort.

The structure of the proposed taxonomy is illustrated in Figure 15, providing a visual representation of the different levels and categories involved. This figure helps contextualize the organization of the taxonomy and provides a reference for understanding how various sounds are classified. By using this framework, researchers and engineers can identify relevant sound events, ensuring that SEC and SED systems are optimized to detect and interpret the most critical sounds in real-world driving scenarios.

3.1.1 Location Level: Internal and External Sounds

At the highest level, the taxonomy distinguishes between internal and external sounds. This distinction is essential, as it reflects how both drivers and SEC and SED systems perceive these sounds. Internal sounds dominate the acoustic

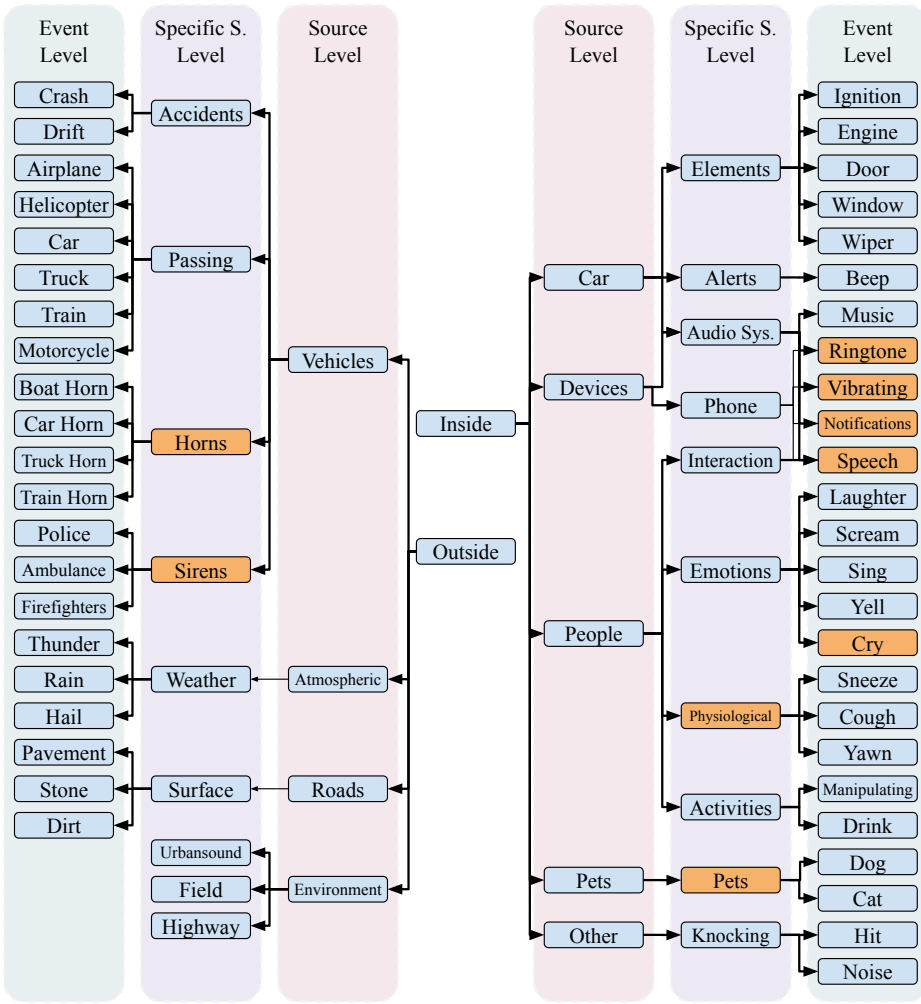


Figure 15: Hierarchical taxonomy of sound events in driving environments.

environment inside the vehicle and, at times, can attenuate the perception of external sounds, such as emergency sirens or traffic noise, which are crucial for safety and attention.

For SEC and SED systems, this distinction is particularly relevant. When capturing sounds from inside the vehicle, the result tends to resemble the occupants' perception: internal sounds dominate, and external sounds may be masked. On the other hand, capturing sounds from outside the vehicle presents new challenges, such as wind noise interfering with the microphone's ability to

record clear audio, as well as the inability to capture internal sounds. Therefore, these systems must take these effects into account, paying attention to the positioning of the microphones and the relative position of sound sources to the vehicle, or implementing strategies that allow them to prioritize certain sound sources while filtering out others.

3.1.2 Source Level: Categorizing by Origin

The source level of sounds is divided into two sublevels: a general categorization that groups sounds by their type of source, and a more specific categorization that refers to concrete acoustic events or situations. This structure helps to better understand where the sounds originate and their role in the auditory environment inside the vehicle.

In the general categorization, Vehicles refers to sounds produced by other vehicles in the surrounding environment. This includes sounds like Horns, Sirens, engine noises from passing vehicles (Passing), and even sounds resulting from collisions or abrupt maneuvers (Accidents). These sounds are essential for perceiving the external environment and detecting potentially hazardous situations.

In contrast, Car focuses on sounds originating from the vehicle under study. This category includes noises generated by the mechanical and structural components (Elements), which are crucial for its operation. Additionally, it separately classifies sounds produced by the car's built-in audio system (Audio Sys.), which, while part of the vehicle, serves a different purpose related to entertainment or navigation. Moreover, Devices covers sounds from other electronic devices inside the vehicle, such as navigation system alerts (Alerts) and mobile phone notifications (Phone), reflecting the increasing role of technology in the driving environment.

Atmospherics includes sounds associated with environmental and weather conditions (Weather). While weather is not typically defined by sound alone, certain situations — like strong wind or hail — provide valuable acoustic information. Weather also influences road conditions, connecting this category

to Roads, which refers to sounds generated by the interaction between tires and the road surface (Surface). These sounds offer insights into pavement quality or the presence of irregularities that may affect driving behavior.

The People category groups sounds produced by human occupants in the vehicle. This includes verbal interactions among passengers, emotional expressions (Emotions), involuntary physiological sounds (Physiological), and even noises arising from activities inside the vehicle, such as adjusting a seat or handling objects (Activities). Separately, Pets accounts for sounds made by animals present in the vehicle, such as barking or meowing, which can influence the cabin's acoustic environment.

Finally, the Other category encompasses sounds that do not fit neatly into the previous classifications. These may include unexpected noises, such as falling objects or knocks inside the vehicle, which, despite lacking a specific category, remain relevant to assessing the overall acoustic environment.

3.1.3 Event Level

The event level represents individual sound events, capturing each occurrence as an independent entity. This level aims to provide a detailed breakdown of sound events, ensuring that all sounds encountered in driving environments are accurately represented. By categorizing sounds with this level of granularity, the taxonomy supports a more precise understanding of the auditory landscape, enabling SEC and SED systems to differentiate between a wide range of acoustic cues — from routine vehicle noises to rare or critical external events. This comprehensive approach ensures that no significant sound is overlooked.

3.2 Acoustic Distractors in Driving Environments

A particularly valuable aspect of this taxonomy is its focus on auditory distractors — sounds that have the potential to divert driver attention, compromising

safety. In Figure 15, these distractors are highlighted in orange, helping to clearly identify and distinguish them within the broader context of sound events. Identifying and highlighting these sounds within the taxonomy supports targeted research into their effects. Common examples include phone alerts, passenger conversations, and sirens from emergency vehicles — all sounds that, despite their importance, can significantly interfere with a driver's focus. The taxonomy, therefore, serves as a foundation for identifying these critical sounds, ensuring they receive appropriate attention in model development and system design.

In the context of driving, acoustic distractors are sounds that divert a driver's attention from the road, increasing the risk of accidents. These distractions are particularly challenging because they often arise unexpectedly and demand immediate attention, even if only for a brief moment. Given that a driver's attention is finite, any interruption can compromise their ability to respond to changes in their environment, potentially leading to dangerous situations.

Having established a comprehensive taxonomy of driving-related sounds, we are now in a position to systematically identify which of these sounds are potential distractors. By analyzing these categories, we can pinpoint those specific sound events that are most likely to divert the driver's attention and impact their ability to focus on driving.

Based on the proposed taxonomy and an extensive review of existing literature, several key categories of acoustic distractors have been identified as most relevant to driving safety. These categories have been selected not only for their prevalence in real-world driving scenarios but also for their potential to significantly impair driver attention and response. Figure 16 provides a graphical representation of these categories, illustrating how different acoustic events can interfere with driving performance and attention.

One of the most common and impactful acoustic distractors is the sound of a phone ringing and vibrating. The ringtone, often accompanied by vibration, can immediately capture the driver's attention, especially if the call is unexpected or perceived as urgent. This distraction is particularly problematic because the psychological response to phone calls—such as the urge to answer or assess the importance of the call—significantly impairs a driver's ability to stay focused on

the road. Numerous studies have shown that phone use, including answering calls, is a major risk factor for accidents, as it diverts visual, cognitive, and manual attention from the task of driving [85, 86, 87, 88]. The risk is amplified in high-traffic situations, where the driver may be under pressure to respond quickly [89]. Similarly, the vibration of a phone can also be distracting, prompting the driver to check their device, further leading to visual and manual engagement that takes focus away from the road. Even if the driver does not answer, the mere action of checking the Phone can delay their reaction times and reduce awareness of potential hazards.

Another significant distractor is the sound of notifications from various apps, such as messaging, social media, or email alerts. These notifications often require the driver to make a decision about whether to check the message, adding a cognitive load that compromises the driver's ability to maintain focus on the road [90, 91]. The frequency and unpredictability of notifications can intensify this distraction, as drivers may feel compelled to assess multiple incoming alerts while simultaneously monitoring their driving environment. This results in a diminished ability to stay fully engaged with the task at hand. Notifications can be especially distracting when they come in quick succession, creating a situation where the driver must constantly divide attention between the road and the phone. This cognitive engagement, even if momentary, can lead to dangerous situations, as it interferes with the driver's ability to respond to unexpected road conditions.

Speech, whether from passengers, voice assistants, or a phone, is another common acoustic distractor that can significantly divert a driver's attention [92, 93, 94]. Conversations with passengers, particularly emotionally charged or complex discussions, require cognitive engagement, making it more difficult for the driver to focus on driving. Similarly, voice interactions with in-vehicle systems, such as navigation or digital assistants, can also draw attention away from the road, especially during critical driving maneuvers [95, 96]. While these systems are designed to be hands-free, the cognitive effort required to process and respond to spoken commands or discussions can still be substantial. This becomes especially problematic when the conversation involves complex or emotionally intense topics, which increase the cognitive load and further diminish attention to the driving

task [97].

The sound of a crying baby can be one of the most distracting acoustic events within a vehicle [98, 99], as it often evokes an immediate emotional response from the driver or passengers. The urgency to attend to the baby, whether by calming the child or determining the cause of distress, can compel the driver to divert attention away from the road. This type of distraction is particularly challenging because it is emotionally charged, making it difficult for the driver to ignore the sound or delay their response. Unlike other distractions that might be more easily dismissed, the crying of a baby demands a rapid response, which can significantly impair a driver's ability to stay focused on navigating the road safely. The emotional intensity of such a distraction makes it difficult for the driver to maintain the level of attention required to respond to dynamic road conditions or hazards.

Physiological sounds, such as sneezing, coughing, or yawning, can also act as distractions [100, 101], although their impact may be less obvious. These natural bodily functions, while generally brief, can momentarily divert the driver's attention, especially if they occur during critical driving maneuvers. Sneezing, for example, can momentarily obstruct the vision of a driver, or the physical act of coughing or yawning can interfere with their ability to respond to changes in the driving environment. While these physiological events are part of normal life, their occurrence during high-risk moments can impair a driver's ability to make quick decisions, leading to potential safety concerns.

Pets in the vehicle, particularly dogs or cats, can contribute to distractions as well. Sounds such as barking or meowing, especially when the pet is restless or requires attention, can lead the driver to respond by checking on the animal. This distraction is compounded by the emotional bond between the driver and their pet, which can make it more difficult to ignore the animal's needs.

Finally, sirens from emergency vehicles — such as police cars, ambulances, or fire trucks — are critical for ensuring safety. However, sirens can also act as significant distractors, especially when the driver is unsure of the direction from which the sound is coming.

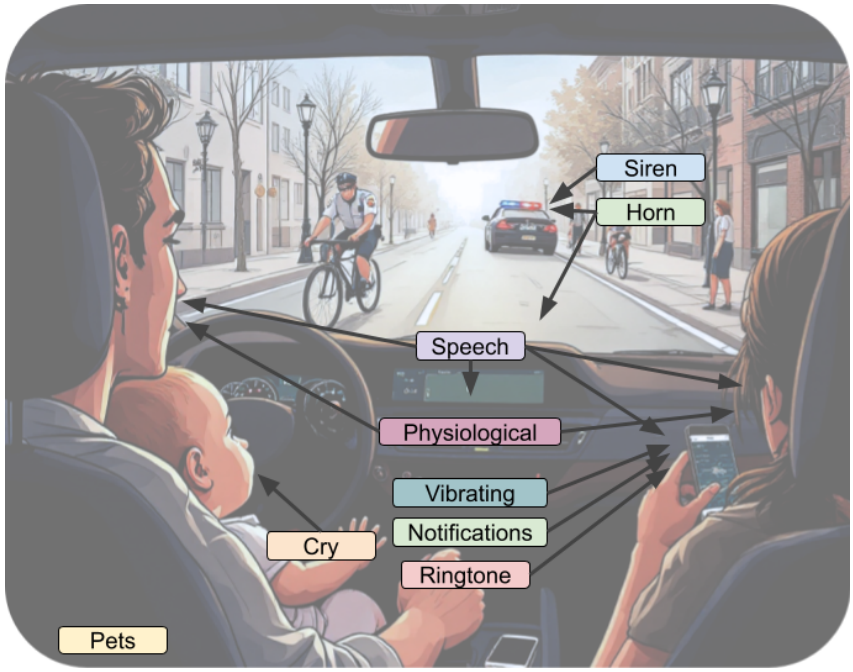


Figure 16: Graphical representation of the key acoustic distractors in the driving environment.

3.3 Development of Acoustic Datasets

The effectiveness of SEC and SED models relies heavily on the availability of accurately labeled datasets. In the context of driving environments, the need for high-quality data is even more critical, as models must differentiate between various acoustic events that can impact driver attention and safety. While most of the sound events described in the previous section can be found in publicly available repositories as isolated samples, existing datasets do not integrate the full range of acoustic distractors in a unified driving scenario.

To address this limitation, we have developed specialized datasets designed to support both SEC and SED tasks. These datasets incorporate a diverse set of labeled sound events, ensuring they reflect real-world driving conditions. Specifically, we introduce two distinct datasets: one focused on acoustic distractors commonly encountered in driving environments and another dedicated

to emotion-related sounds that can influence a driver's cognitive state and reaction times. By aligning these datasets with the taxonomy, we ensure comprehensive coverage of critical auditory events for improved model training and evaluation.

The following subsections describe the methodology used to construct these datasets, detailing their structure, data sources, and key characteristics.

3.3.1 Datasets of Acoustic Distractors in Driving

Building on the taxonomy of driving-related acoustic events, we developed a dataset specifically tailored for the detection and classification of auditory distractors (see Section 3.2). These distractors, previously identified as critical factors affecting driver attention, are now integrated into a structured dataset designed for SED and SEC tasks. While the taxonomy encompasses a broad range of sound events, this dataset focuses on those most relevant to driving distractions. Additionally, the remaining sound events, not explicitly categorized as distractors, help to create a more realistic driving scenario in the synthetic dataset, simulating the full range of sounds a driver might encounter and ensuring comprehensive coverage of real-world scenarios.

Two datasets of the data have been collected for SED task: one systematic and one real. To build the synthetic dataset, as detailed in the Section 2.3.1 on synthetic data generation, isolated sound events were obtained from publicly available repositories with unrestricted usage rights [102, 103, 104, 105, 106, 107, 108, 109, 110]. These samples were subjected to rigorous quality control to verify their authenticity and acoustic clarity, discarding recordings with excessive noise or missing target events. The selected sounds were used to generate 10-second audio mixes using the Scaper library [111], allowing for precise event placement and random combinations reflecting diverse driving conditions. Data augmentation techniques, such as time stretching and pitch shifting, were applied to improve the variability and generalization of the model (see Section 2.3.3).

To simulate realistic in-vehicle acoustic conditions, all generated mixtures incorporate background noise extracted from real car recordings [106, 107].

This background noise was randomly selected to represent different driving environments, and events were introduced with signal-to-noise ratios between -20 dB and -10 dB to balance event prominence. Additionally, real-world impulse responses measured within car cabins were applied to accurately model in-car sound propagation, such as speech from passengers or vehicle audio output [112].

The final dataset comprises 19,000 audio files (approximately 53 hours), partitioned into training (15,000 samples), validation (2,000 samples), and testing (2,000 samples) subsets. These subsets are referred to as `SED_SYN_Train`, `SED_SYN_Val`, and `SED_SYN_Test`, where `SED` denotes the dataset's intended use for sound event detection tasks, and `SYN` indicates its synthetic nature. Isolated sound events were split in the same proportion to prevent data leakage and ensure unbiased evaluation. The dataset provides hard labels indicating event classes and temporal boundaries, formatted for flexibility across various SED and SEC applications.

Table 1 provides a detailed breakdown of the dataset, presenting the number of isolated events and their respective appearances in the training, validation, and testing subsets. The number of isolated samples for each event depends on the ease or difficulty of acquiring relevant audio recordings, but each event has a minimum of 50 samples. To ensure balance, the number of appearances is roughly the same across event classes; however, distractor events, identified as critical for driver attention, are more frequently represented in the dataset to reflect their higher relevance for the task.

The labels for the dataset are provided in two formats: a full set of event-level annotations with 41 classes, and a simplified set of labels indicating only the 9 main distractor events. For the latter, some events have been combined based on the taxonomy to reflect their common characteristics. For example, the Siren category includes both ambulance and police sirens, as detecting the specific source is less critical for identifying the distraction. Similarly, events like Horn, Pets, and Physiological sounds have been merged into broader categories, with the primary focus on their role as distractors rather than their exact origin.

The real dataset, presented exclusively for testing purposes, consists of three distinct scenarios designed to evaluate the performance of models trained on

Table 1: Distribution of isolated event samples and their corresponding appearances in the `SED_SYN_Train`, `SED_SYN_Val`, and `SED_SYN_Test` subsets.

Event	Isolated				Appearances			
	Total	Train	Val	Test	Total	Train	Val	Test
Airplane	271	217	27	27	708	560	76	72
Ambulance	50	40	5	5	2991	2341	309	341
BoatHorn	50	40	5	5	2922	2305	314	303
Car	79	63	8	8	651	516	72	63
CarHorn	232	186	23	23	2966	2364	272	330
Cat	175	140	18	18	2857	2247	311	299
Cough	66	53	7	7	2957	2342	311	304
Crash	50	40	5	5	706	540	90	76
Cry	545	436	55	55	2950	2326	295	329
Dog	295	236	30	30	2986	2380	307	299
Door	50	40	5	5	697	550	79	68
Drift	50	40	5	5	687	533	83	71
Drink	50	40	5	5	735	584	75	76
Firefighters	50	40	5	5	2910	2297	297	316
Hail	64	51	6	6	695	546	74	75
Helicopter	108	86	11	11	650	508	75	67
Hit	50	40	5	5	687	538	76	73
Ignition	50	40	5	5	685	552	71	62
Laughter	123	98	12	12	716	571	79	66
ManipulatingObjects	83	66	8	8	654	517	62	75
MicrophoneNoise	50	40	5	5	663	529	65	69
Motorcyle	50	40	5	5	724	564	87	73
Music	3986	3189	399	399	636	505	67	64
Notifications	50	40	5	5	2862	2246	305	311
Police	50	40	5	5	2957	2323	317	317
Rain	107	86	11	11	698	542	80	76
RingTone	399	319	40	40	2835	2246	312	277
Scream	50	40	5	5	697	548	79	70
Sing	139	111	14	14	681	530	77	74
Sneeze	53	42	5	5	2882	2281	287	314
Speech	266	213	27	27	3020	2356	328	336
Stone	87	70	9	9	705	565	68	72
Thunder	50	40	5	5	695	555	62	78
Train	97	78	10	10	2801	2220	260	321
TrainHorn	50	40	5	5	715	553	82	80
Truck	50	40	5	5	3004	2343	340	321
TruckHorn	50	40	5	5	2988	2387	280	321
Vibrating	50	40	5	5	706	571	74	61
Window	50	40	5	5	697	544	68	85
Wiper	56	45	6	6	648	518	60	70
Yawn	50	40	5	5	2861	2286	282	293

the synthetic dataset under realistic conditions. The subsets are referred to as SED_REAL_S1, SED_REAL_S2, and SED_REAL_S3, where REAL indicates the real-world dataset, and the suffixes S1, S2, and S3 refer to the specific scenarios captured during the testing phase. The main challenge in creating this subset lies in the complexity of real-world data collection, where factors such as environmental noise, the diversity of events, and the variability in the occurrence of distractors affect the quality and consistency of the data. As reflected in Table 2, the number of occurrences of events varies across the three scenarios due to these complexities. In addition, while some events are well-represented, others are less frequent or absent in certain scenarios, which may impact the evaluation of the models.

In SED_REAL_S1, audio was recorded during an actual car trip, where specific distractor events were either naturally occurring or deliberately provoked to test the models. This scenario resulted in 684 samples (11.4 minutes of audio), for 10-second clips with a 1-second sliding window. This scenario features annotations for five out of the nine distractor events (Speech, Horn, Physiological, Ringtone, and Notification), and while the subset is relatively small compared to the synthetic one, it serves to assess how well models generalize from synthetic to real-world data.

SED_REAL_S2 subset, which consists of 475 samples (79.2 minutes of audio), expands the real-world testing by incorporating a greater variety of event combinations. Here, each 10-second recording contains up to four events, with some segments including overlapping classes. The recordings were made with a high-quality microphone inside the vehicle, with the driver and passengers creating and interacting with various distractor events. For this scenario, 8 out of the 9 classes are represented, leaving out only Vibrating, and no overlapping windows are used. This explains why, despite having a longer duration, the number of samples is lower than in Scenario 1.

Finally, SED_REAL_S3 was designed to capture data under different placement conditions, using multiple microphones inside and outside the vehicle to record both external sounds. The total duration of this scenario was 35.4 minutes, and the subset includes 2,122 samples obtained using a 1-second sliding window over 4-second segments. Only two types of events were labeled in this dataset: Horns

Table 2: Occurrences of distractor events across three real-world driving scenarios.

	SED_REAL_S1	SED_REAL_S2	SED_REAL_S3
Cry	0	12	0
Horn	24	4	52
Notifications	19	31	0
Pets	0	7	0
Physiological	27	4	0
RingTone	71	20	0
Siren	0	4	273
Speech	806	644	0

and Siren. This subset is constituted by two parts: `SED_REAL_S3_CleanCabin` and `SED_REAL_S3_NoisyCabin`. The former contains recordings without internal noise sources, while the latter includes internal sounds (e.g., conversations, radio), allowing us to evaluate the degree of masking that internal noise introduces to the detection of external events.

In addition to the previously described differences between the `SED_REAL_S1`, `SED_REAL_S2`, and `SED_REAL_S3` scenarios, there are also variations in the microphones and recording setups used. In `SED_REAL_S1`, a generic lapel microphone (mono) is used, connected to a Raspberry Pi 4, which serves as the recording device. For `SED_REAL_S2`, a Zoom H6 recorder with XYH-6 microphones (stereo, 2 channels) is employed. A similar configuration is used in `SED_REAL_S3`, but with the addition of DIY WM-61A microphones (mono) to capture sound from the exterior environment.

In all cases, the interior microphones are centrally positioned within the recording environment, while the exterior microphones are distributed across the four points marked in red in Figure 17. It is important to note that, for the construction of this dataset, all recordings are stored in a single channel. For exterior recordings, the channel corresponding to the upper-right microphone is used, while for interior recordings, the left channel is always selected.

The complete access to the distractor dataset, which includes all subsets, is available at the following Kaggle link: https://www.kaggle.com/datasets/ccastorena/sound-event-detection-for-driver-safety.

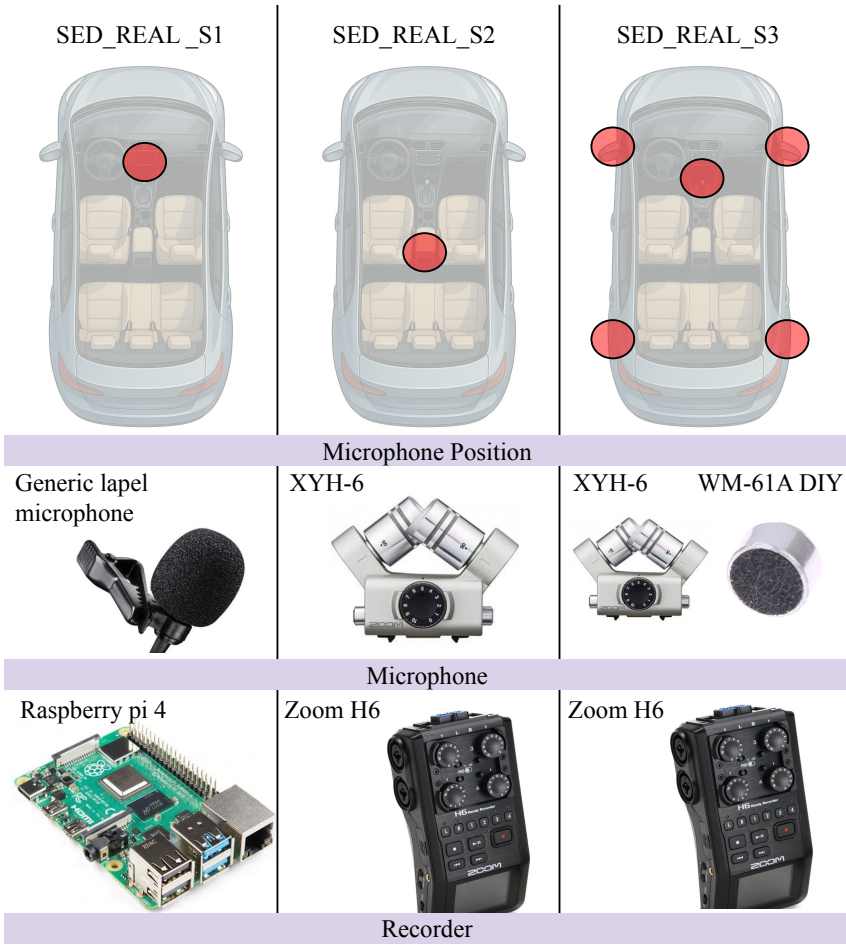
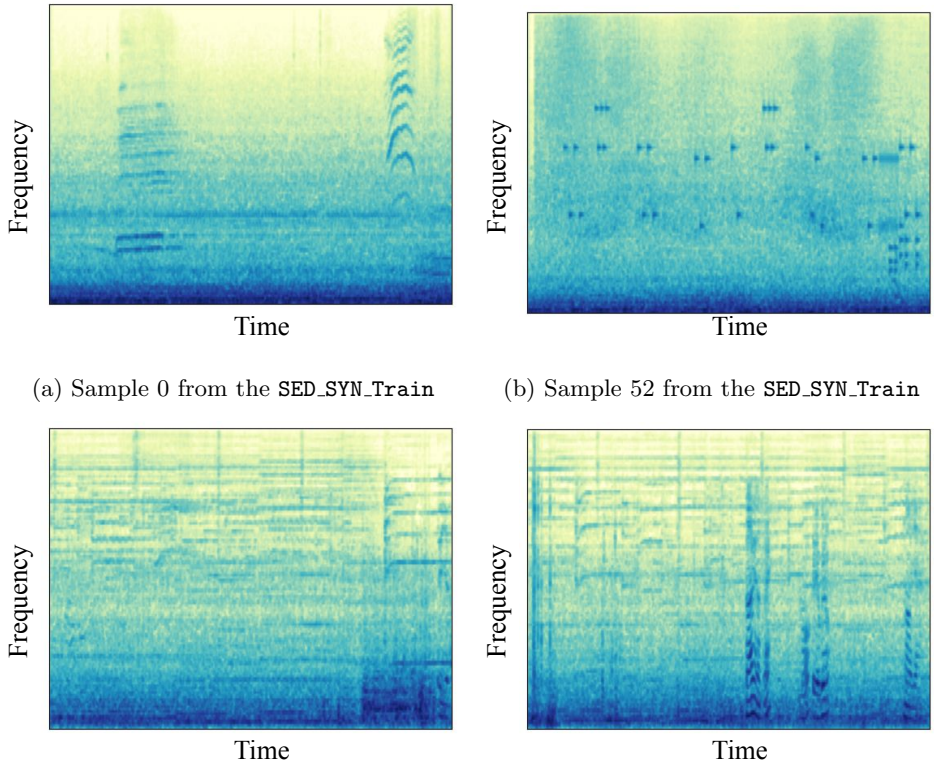


Figure 17: Microphone placement for interior and exterior recordings in the real dataset.

To facilitate reproducibility and maintain consistency with the experiments conducted in this thesis, we suggest using a Mel spectrogram as the default audio representation for the provided datasets. Specifically, we recommend the configuration described in Section 2.2.1, where the Mel spectrogram is computed using 128 Mel bands, a window length and FFT size of 2048 points, a Hamming window function, and a hop size of 256. With a sampling rate of 16,000 Hz, this configuration covers the frequency range from 0 to 8,000 Hz, offering a detailed spectral representation over time. The resulting spectrogram dimensions are 626



(a) Sample 0 from the `SED_SYN_Train`

(b) Sample 52 from the `SED_SYN_Train`

(c) Sample 21 from `SED_REAL_S1`

(d) Sample 40 from `SED_REAL_S1`

Figure 18: Mel spectrogram examples for different samples from the synthetic and real datasets.

frames, 128 Mel bands, and one channel per audio sample.

However, as these datasets are distributed in raw audio format, users are free to explore alternative acoustic representations depending on the specific needs of their models or target applications.

Although these representations are not directly provided in the dataset, the information outlined here allows for their reproducible computation. Figure 18 presents examples of Mel spectrograms for events in both the synthetic and real datasets, highlighting the spectral differences and variability between the two subsets.

3.3.2 Datasets of Emotion-Related Sounds in Driving

As discussed in Section 3.2, emotional state can significantly influence a driver's cognitive and behavioral responses, potentially impacting safety even if emotions are not traditionally categorized as distractors. High emotional load—such as stress, anger, or sadness—can impair attention, reaction time, and decision-making. Therefore, understanding the emotional content of speech is relevant in driving contexts where subtle changes in tone or affect may signal risk.

To explore this dimension, we introduce a dataset specifically designed for the task of speech emotion recognition (SER), which can be understood as a specific instance of SEC, where the goal is to classify the emotional state conveyed through spoken language. Although SER is not, strictly speaking, a SED task, it becomes compatible with the SED framework when combined with speech activity detection models—such as those trained on the distractor dataset presented earlier. By first identifying speech segments, a system can subsequently classify the emotional state expressed within them, effectively segmenting and labeling emotional events in time.

Unlike the synthetic approach used in the previous dataset, this resource is built by consolidating several established SER datasets without artificially inserting events in time. Its main objective is to support the development of models capable of generalizing to new speakers, particularly by addressing the challenge of speaker adaptation. This allows researchers to move beyond emotion classification in isolated settings and towards systems that can operate under more realistic, dynamic conditions.

To construct this resource, a diverse group of publicly available emotional speech datasets have been selected, speaker demographics, and emotional categories, providing a rich foundation for training and evaluation:

- EMO-DB (Berlin emotional speech database) [113]: This dataset contains 800 German recordings, expressing seven different emotions: anger,

boredom, disgust, fear, happiness, sadness, and neutral. It provides valuable data for training models that can recognize emotions in speech.

- MESD (Mexican emotional speech database) [114]: A Spanish dataset with 864 samples, categorized into six different emotions: anger, disgust, fear, happiness, sadness, and surprise. MESD offers a culturally and linguistically distinct set of data for emotion recognition.

- RAVDESS (Ryerson audio-visual database of emotional speech and song) [115]: This dataset features 7,356 recordings from 24 native English actors. It includes a range of emotions and is often used in training models for emotion classification tasks.

- ESD (Emotional speech dataset) [116]: A multilingual dataset containing 7,000 dialogues in both English and Mandarin, expressing five core emotions: anger, happiness, sadness, surprise, and neutral. This dataset provides an excellent opportunity to explore emotion recognition across different languages and cultural contexts.

Since each of these individual datasets has its own set of emotion labels, we have limited ourselves to maintaining only the samples for the following emotions: anger, happiness, sadness, surprise, and neutral. Additionally, we have preserved the speaker information to avoid speaker duplication during the training and testing process. It is important to highlight the ESD dataset, as it integrates the largest number of speakers and emotions per speaker. This feature makes ESD a key dataset in terms of variability and real-world application for SER systems.

To enable experiments on speaker adaptation, a dataset comprising 37 speakers was constructed by combining EMO-DB, MESD, and RAVDESS. Given that emotional expression varies significantly between individuals—even within the same language or region—this resource provides a basis for developing strategies that generalize across speakers. This subset is designated as the source domain (SD), with the corresponding training and testing splits referred to as `SER_SD_Train` and `SER_SD_Test`, respectively. These subsets are intended to support the training of models capable of handling speaker variability.

Table 3: Overview of the SER_SD and SER_TD subsets for speech emotion recognition.

	SER_SD			SER_TD
	EMO-DB	MESD	RAVDESS	ESD
Angry	144	196	192	7000
Happy	144	71	192	7000
Sad	144	62	192	7000
Surprise	144	0	192	7000
Neutral	144	79	96	7000
All	576	408	864	
		1848		35000
Speakers	3	10	24	20

To evaluate the generalization capabilities of models to unseen speakers, the ESD dataset is employed as the target domain (TD). ESD is particularly valuable in this context due to its comprehensive coverage and speaker diversity. The corresponding subsets are labeled `SER_TD_Train` and `SER_TD_Test` for training and testing, respectively. The characteristics of both the source and target domains are summarized in Table 3, supporting the evaluation of speaker adaptation strategies under realistic cross-speaker conditions.

Unlike the SED dataset, which provides raw audio signals, this dataset is distributed in the form of high-level representations extracted using a transformer-based model. Specifically, each audio segment is encoded using a fine-tuned version of Wav2Vec [14], adapted for emotion recognition as described in Wagner et al. [15]. This model combines convolutional layers—which operate over the Mel spectrogram representation (see Section 2.2.1 and 2.2.2)—with a stack of transformer layers that produce a time-distributed output (see Section 2.2.2). From the last transformer layer, a global average pooling operation is applied to obtain a fixed-length embedding for each sample.

During fine-tuning, using the MSP-Podcast [117] and IEMOCAP [118] datasets, the convolutional layers of Wav2Vec are frozen while the transformer layers are updated. This approach effectively captures both low-level acoustic details and high-level semantic information relevant to emotion classification.

These representations, 1024-dimensional in size, are provided directly to

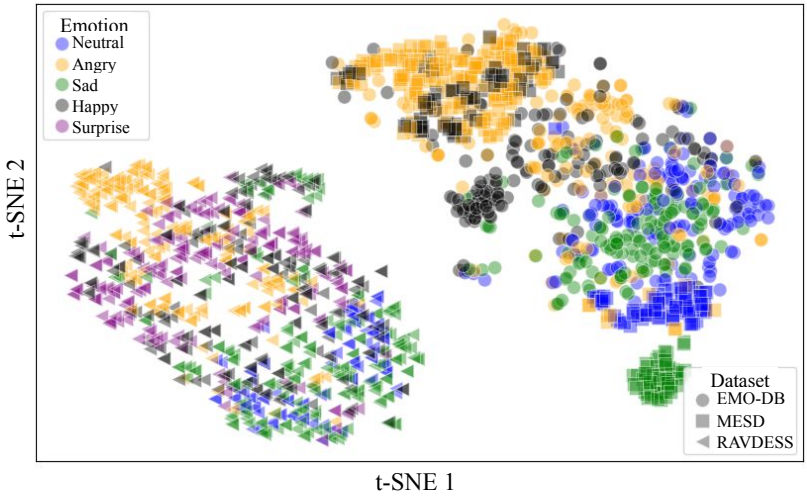


Figure 19: t-SNE visualization of the 1024-dimensional transformer-based representations for the 37 speakers and 5 emotions in the SD subset.

facilitate experiments focused on speaker adaptation and emotional classification, eliminating the need for raw audio processing. Additionally, for visualization purposes, a 2D projection of these embeddings was computed using t-SNE. Figure 19 shows the emotional distribution of 1,848 samples from the `SER_SD` subset, where each emotion is color-coded and each individual dataset is distinguished by shape. Although the t-SNE projection has known limitations, the figure illustrates how emotions and individual datasets cluster differently, underscoring the importance of speaker-aware modeling in emotion recognition tasks.

The 1024-dimensional transformer-based representations for both the SD and TD subsets are provided and can be accessed through the following link: https://www.kaggle.com/datasets/ccastorena/emotion-recognition-dataset-for-speaker-adaptation.

3.4 Strengths, Limitations and Applications of the Framework

The proposed framework presents several strengths that make it a valuable contribution to the field of SEC and SED for driving environments. One of the most significant advantages is the creation of a specialized taxonomy tailored to the specific needs of driving scenarios. Unlike previous work, which often relies on general-purpose event detection models, this framework focuses on a wide range of sound events that are particularly relevant to drivers, including not only common distractors such as phone alerts and sirens but also more driving-specific events like engine sounds, tire-road interactions, and even potential accidents.

The datasets introduced—both synthetic and real—serve as a crucial foundation for training and evaluating models in this context. The `SED_SYN` dataset, while synthetic, provides a controlled environment for experimenting with a variety of sound events under consistent conditions. This dataset allows for fine-tuned model training, ensuring that the detection algorithms are robust to different types of sounds and temporal variations. The `SED_REAL` dataset, on the other hand, introduces real-world variability, offering an invaluable resource for testing the performance of models under unpredictable, real-life conditions. By combining these two datasets, the framework enables the development of models that are both precise and adaptable.

The practical applications of this framework are vast. In the automotive industry, the ability to accurately detect and classify sound events could enhance advanced driver-assistance systems, contributing to the development of safer vehicles. For example, a model trained using the `SED_SYN` dataset could be deployed to identify potential hazards such as approaching emergency vehicles or road accidents. Meanwhile, the `SED_REAL` dataset would allow these models to be tested and refined for real-world accuracy, accounting for the diverse and unpredictable nature of sound events in everyday driving scenarios.

Beyond driving safety, this framework could also be adapted for use in other domains, such as in the design of noise-canceling systems, speech recognition

systems in noisy environments, or even virtual assistants that need to differentiate between relevant and irrelevant sounds. By focusing on sound events that directly affect human perception and behavior, the framework has the potential to significantly impact multiple fields where sound recognition plays a crucial role.

Although we have made an effort to design a comprehensive taxonomy for SED in the context of driving, it is important to acknowledge that not all relevant events may be covered. While an extensive range of events influencing driving has been included, there will always be less frequent, but still important, events that may not be adequately represented. The limitation of a rigid taxonomy is that it is not always possible to predict all sounds that could be relevant to the driving context, and some emerging events might not have been anticipated in its design.

Additionally, one of the inherent challenges in the taxonomy is the subjectivity in classifying certain events as distractors. Depending on the driver and their experiences, the perception of a sound event as a distractor can vary significantly. For example, what might be a significant distraction for one person, such as a phone call, may not be as distracting for another. This subjectivity can influence the effectiveness of models in real-world situations, as what could be a critical distractor for one driver may not have the same impact on another.

Regarding the datasets used, both the `SED_SYN` and `SED_REAL` datasets provide a solid foundation for training and evaluating models in SED tasks but come with some limitations. In the case of the `SED_SYN` dataset, one of the main issues is the representation of certain events, such as *Crash*, which are underrepresented due to the difficulty of obtaining isolated recordings of these events. This is an inherent challenge in synthetic data collection, as events like accidents are rare and difficult to obtain without compromising their quality and temporal boundaries.

On the other hand, the `SED_REAL` dataset, being collected in real-world scenarios, faces the limitation of the natural variability of events. In these cases, not all expected events occur predictably, leading to inconsistent representation within the dataset. This is because the occurrence of certain sound events depends on unpredictable situations, such as the presence of emergency vehicles or accidents. Additionally, the frequency of event occurrence in the synthetic dataset does not always align with what is observed in real-world environments,

which may affect the model's ability to generalize in unseen conditions during training.

Another relevant limitation is the nature of the `SED` dataset and its impact on validation. Since it is based on publicly available sources, biases are likely to emerge when pre-trained models are used. These models may be influenced by the same data sources used during their training, which could compromise the independence of validation and, consequently, the robustness of the results obtained. This suggests that model validity should be carefully assessed to ensure that results are not contaminated by these biases.

Finally, the `SER` dataset, used for speaker adaptation experiments in emotion recognition tasks, presents an inherent limitation in the nature of the recorded emotions. The emotions in this dataset are acted out, which may not accurately reflect the spontaneous emotional responses of individuals in real-world situations. This distinction can be crucial, as genuine emotions tend to be more complex and variable, which could affect the model's applicability in real-world scenarios, where emotional responses are not always as predictable or easy to classify.

In summary, while the datasets used in this framework provide a valuable foundation for the development of SEC y SED models in driving scenarios, they present several limitations that should be considered. These limitations, ranging from event representation in the taxonomy to challenges in real-world data collection and the nature of recorded emotions, pave the way for future improvements and adjustments. Addressing these challenges will help enhance the accuracy and applicability of the proposed framework in real-world conditions.

Chapter 4:
CNN-Based Approach for Efficient Sound Event Detection

This chapter presents another significant contribution of this thesis: a novel approach for sound event detection that leverages convolutional neural networks. Specifically, we explore how CNNs can be applied to audio data for efficient and accurate sound event detection in real-world environments, with a particular focus on safety-critical applications.

The foundation of this approach is the adaptation of a popular object detection model, YOLO (You Only Look Once), originally designed for image-based tasks, to handle audio data. While this approach is not the first of its kind, such as the YOHO (You Only Hear Once) model, throughout this chapter we highlight the main differences and discuss other state-of-the-art techniques applied to sound event detection. By transforming audio signals into spectrograms, YOLO is able to detect and locate sound events in the time-frequency space, providing real-time classification and detection of various sound events. This approach applies the power of deep learning models designed for vision tasks to the realm of sound, enabling effective detection of diverse acoustic events.

The chapter also discusses the methodology behind the adaptation process, detailing how YOLO is adjusted for audio-based sound event detection. We explore the experimental setup, including the model architecture and evaluation metrics used to assess performance. By focusing on both detection accuracy and inference time, we demonstrate how this CNN-based approach balances high performance with efficiency, a crucial requirement for practical SED applications.

Furthermore, we present the results of our experiments, highlighting how this CNN-based approach outperforms traditional methods in terms of accuracy, robustness, and processing time. We also analyze the challenges encountered during the adaptation of YOLO to audio data and how they were addressed.

Through the development of this CNN-based approach, this chapter contributes to the advancement of SED systems, providing an efficient and scalable solution for SED in real-world driving environments. The results and information presented in this chapter can help create more effective systems for sound event detection, ultimately improving applications in critical safety areas such as autonomous driving, surveillance, and emergency response.

4.1 Current Strategies to Sound Event Detection

SED has greatly benefited from the advancements in deep learning techniques, particularly those involving CNNs and RNNs. These architectures have been shown to be highly effective for extracting and classifying complex patterns in time-series audio data. The integration of CNNs and RNNs enables the detection of both local (frequency domain) and temporal features of audio, which are crucial for accurately identifying and locating sound events within audio signals.

In the following subsections, we will review specific neural network architectures that have been widely adopted for SED tasks: CRNNs, YOHO, and our proposed approach YOLO for audio data. Each of these models brings unique strengths to the problem of SED, and we will explore their differences, advantages, and applications in more detail.

4.1.1 CRNN Model for Sound Event Detection

CRNNs have emerged as a prominent architecture for SED. These models integrate convolutional layers, responsible for extracting time-frequency features from spectrograms, with recurrent layers that capture temporal dependencies in audio signals.

The convolutional layers apply a series of trainable filters to the spectrogram, detecting characteristic time-frequency patterns associated with sound events. The resulting feature maps retain the input's temporal and spectral relationships,

which are then processed by recurrent layers—commonly GRUs—to model the progression of events over time (see Section 2.2.2).

The final stage consists of fully connected layers that classify the extracted features, mapping them to probabilities for different sound event classes (see Section 2.2.3).

Over time, various CRNN adaptations have been proposed, keeping a similar structural foundation while integrating techniques for improved regularization. One such method is the exponential moving average (EMA) function (see Section 2.3.3), utilized in semi-supervised learning through a mean teacher framework [76]. Additionally, CRNN models often incorporate both hard and weak labels [119], a widely used approach in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge [120].

More recent CRNN approaches leverage pre-trained models to enhance detection accuracy [121, 122]. However, this strategy may compromise the independence of synthetic test data, as pre-trained models often rely on open datasets that might include overlapping sources. Furthermore, these models typically have a high parameter count to represent a diverse range of sound events, which can adversely impact real-time performance—a crucial factor in our study [123, 124].

One key consideration in CRNNs is the recurrent component, which, while essential for capturing signal dynamics, can introduce significant computational costs, leading to increased inference times.

The reference baseline for the DCASE 2021–2023 challenge [125, 29], used as the primary state-of-the-art benchmark for our proposal, is illustrated in Figure 20. This architecture consists of seven convolutional layers (see Section 2.2.2). The extracted feature maps are reshaped and processed by two Bi-GRU layers (see Section 2.2.2), which, with the help of a series of dense layers, predict the presence or absence of events at each time step.

Figure 20 presents the detailed CRNN architecture. The model is composed of multiple convolutional blocks, each with a 3×3 kernel. The number of filters per layer follows this sequence: 16, 32, 64, 128, 128, 128, 128. Each block also integrates batch normalization, a ϕ_{GLU} activation function, and pooling

operations with sizes of 2×2, 2×2, 1×2, 1×2, 1×2, 1×2. These pooling layers progressively reduce the input size, resulting in an output shape of [B, 128, 156, 1], where B represents the batch size.

To meet the input requirements of the recurrent layers, the feature maps are transposed and the extra dimension is removed. The Bi-GRU layer then generates an output of shape [B, 156, 256], with 128 hidden units per direction. The final dense layer produces event activation predictions over 156 time frames for C event classes. Overall, the architecture comprises approximately 1.1 million trainable parameters.

The input to the CRNN model is a Mel spectrogram computed from the audio mixtures. This spectrogram is generated using the configurations mentioned in Section 3.3 and serves as input to the CRNN model. The resulting spectrogram, with dimensions of 626 time frames, 128 Mel bands, and one channel, serves as the input to the CRNN model.

Regarding event labeling, the CRNN follows a frame-wise hard labeling strategy, segmenting time into discrete frames where each frame is assigned a binary label: 1 for frames containing the event and 0 for those without (see Section 2.2.3). This results in a matrix shaped according to the number of frames and the number of events. In this study, the model produces 156 frames corresponding to nine distinct sound events. Additionally, weak labels are automatically derived by determining whether an event occurs anywhere within a given segment.

To enhance model performance, several training strategies are employed. A key approach is mix-up augmentation (see Section 2.3.3), a widely used technique in machine learning to improve generalization. Mix-up involves linearly interpolating two randomly selected training samples—both their spectrograms and labels—to create synthetic data points, thereby increasing dataset diversity and reducing overfitting.

To align with the official DCASE challenge setup, the CRNN model is trained for 200 epochs using a mini-batch size of 48 and optimized with Adam. The initial learning rate is set to 0.001, following an exponential warm-up schedule. Model selection is based on the highest validation performance. The training

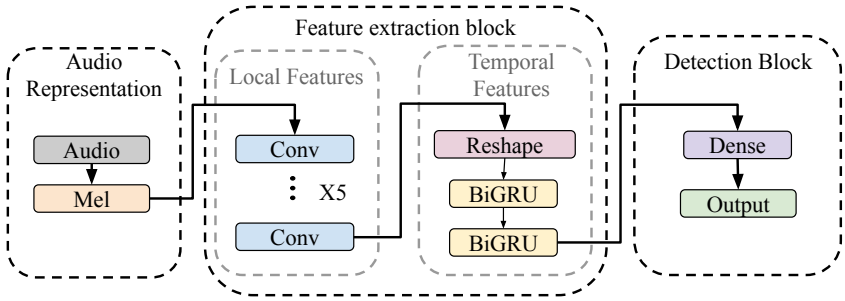


Figure 20: Architecture of the CRNN baseline model used in the DCASE 2021–2023 challenge.

process evaluates three types of error: (1) Hard error, computed using binary cross-entropy ($\mathcal{L}_{BCE}$) loss to assess classification accuracy against hard labels (see Section 2.2.4); (2) Weak error, measuring the model's ability to recognize events without precise timing annotations, also utilizing $\mathcal{L}_{BCE}$ loss; (3) Consistency error, applied to unlabeled data, which computes the mean square error ($\mathcal{L}_{MSE}$, see Section 2.2.4) between the student model's predictions and an exponentially updated teacher model, enforcing stability and consistency in predictions even when labeled data is unavailable. The weak and hard error components are each weighted with a factor of 1 during training, while the consistency error is assigned a weight of 2 to emphasize its importance in the semi-supervised setting.

4.1.2 YOLO-based Model for Sound Event Detection

In this section, we explore the application of YOLO-based architectures for SED, extending the principles of object detection to the audio domain. Although prior studies have adapted YOLO for SED [126, 127], most efforts have focused on earlier versions of the model, particularly through the YOHO (You Only Hear Once) framework.

YOHO reformulates SED as a regression task, predicting the temporal boundaries—i.e., onset and offset times—of sound events, rather than relying

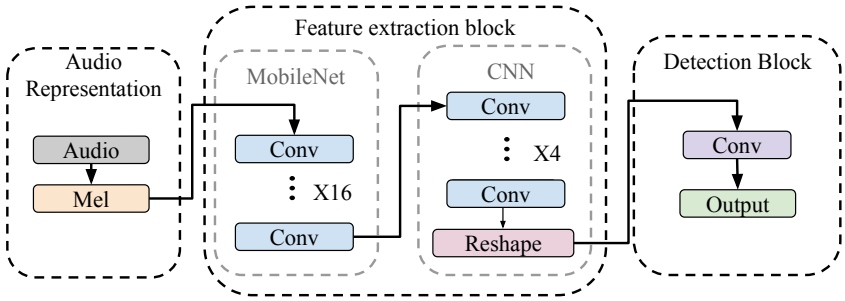


Figure 21: Architecture of the YOHO model based on a MobileNet and custom CNN.

on traditional frame-wise classification.

The YOHO architecture builds upon MobileNet [128], a lightweight and efficient convolutional neural network used as the feature extraction backbone. In addition, the model incorporates a custom CNN block designed both to adapt to the characteristics of the VOICe dataset [129] and to produce a fixed output shape of 9×9. This corresponds to 9 temporal segments, each containing predictions for 3 sound event classes. For each class, the model outputs a probability along with onset and offset estimates, resulting in 9 values per time step.

YOHO employs depthwise separable convolutions throughout its architecture. Unlike standard convolutions, this approach factorizes the operation into two stages: a depthwise convolution that applies a single filter per input channel, followed by a pointwise (1×1) convolution that combines the outputs across channels.

The model operates on input Mel spectrograms of size 801×64, capturing both spectral and temporal characteristics of the audio signal. A series of convolutional layers with stride 2 progressively reduces the temporal resolution, enabling predictions of class probability, onset, and offset times for each event. This structure also allows YOHO to detect overlapping (polyphonic) events, such as simultaneous speech and background music.

The model (illustrated in Figure 21) is trained using a custom loss function that incorporates $\mathcal{L}_{\text{MSE}}$, jointly optimizing classification performance

and temporal boundary precision.

Although YOHO remains an early-stage approach, it has shown moderate improvements over CRNN-based models, particularly in its ability to detect and localize events more accurately by leveraging YOLO-inspired design principles. These improvements are especially notable when compared to conventional CNN-based architectures, which lack the object detection mechanisms integrated into YOLO [45, 127].

Building upon the principles demonstrated by YOHO, this work proposes a more advanced YOLO-based architecture for SED, inspired by the design of YOLOv5 [130], which achieves results comparable to more recent YOLO variants while remaining computationally efficient and easier to integrate into low-latency systems (see Appendix B). While YOHO adapts early YOLO versions with a lightweight MobileNet backbone and a custom output block, our approach incorporates the architectural advancements of YOLOv5, including its modular design and improved feature representation.

As illustrated in Figure 22, the proposed architecture consists of three key components: the Backbone, which extracts hierarchical audio features from the input spectrogram using stacked convolutional layers; the Neck, based on a Feature Pyramid Network (FPN), which enables multi-scale feature fusion to improve detection across events of varying temporal lengths; and the Detection Head, which outputs event classes along with their temporal positions and durations across multiple feature scales.

A key difference in our formulation is the representation of temporal boundaries. Instead of directly regressing onset and offset times, our model predicts the midpoint and duration of each event. This reformulation simplifies the regression task, improves robustness to boundary noise, and yields more stable localization outputs.

We evaluate three YOLOv5 variants—YOLOv5n, YOLOv5s, and YOLOv5m— which differ in depth and width, resulting in different model sizes and capacities, with approximately 1.9 million, 7.2 million, and 20 million parameters, respectively. Despite these differences, all variants share the same architectural structure, allowing for scalable deployment under varying

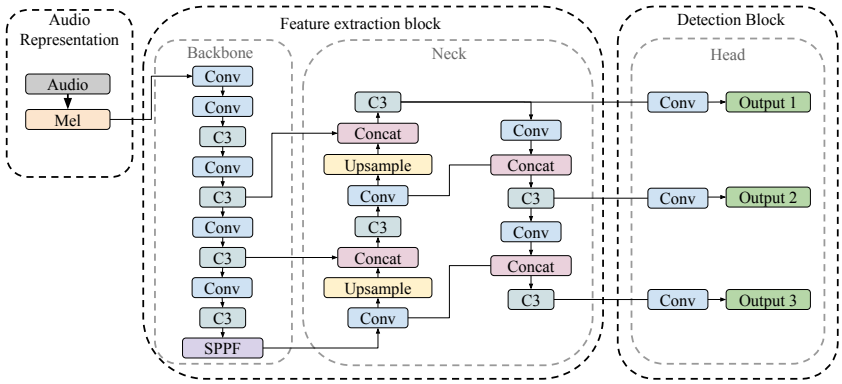


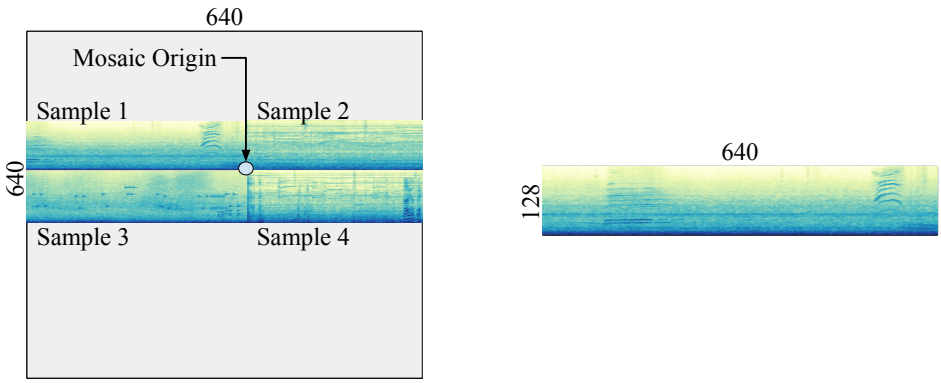
Figure 22: Architecture of the proposed YOLO-based architecture for SED.

computational constraints.

In contrast to CRNN-based models, which rely on recurrent layers to model temporal dynamics, our architecture remains fully convolutional. This design choice allows for highly parallel computation and significantly reduces inference latency—making it especially well-suited for real-time and embedded SED applications.

The input to the YOLO-based model is a Mel spectrogram, consistent with the representation used in the CRNN architecture. However, to conform to the fixed input dimensions required by the YOLOv5 framework, the spectrogram is adjusted to a standard length of 640 time frames. Given that the original spectrogram has 626 frames, we apply zero-padding by adding 7 frames at the beginning and 7 frames at the end, resulting in the desired input size of 640 frames. This padding ensures compatibility with the model's stride and facilitates consistent feature extraction across samples.

During training, a data augmentation technique known as mosaic augmentation [131] is applied, which transforms the input into a 640×640 canvas by randomly combining four spectrogram samples. These samples are placed at random positions within the canvas, ensuring that they are always aligned next to each other without overlapping, as illustrated in Figure 23a. It is important to note that, due to the random positioning along the temporal axis during mosaic augmentation, individual samples become partially truncated. However, along the



(a) Mosaic augmentation applied during training

(b) Standard input format during inference

Figure 23: Illustration of the mosaic data augmentation technique during training and the standard input format during inference.

frequency axis, the full range of 128 Mel bands is preserved to maintain consistent spectral representation. This augmentation technique is crucial during training, as demonstrated by our preliminary experiments detailed in Appendix A. For inference, the model receives the original input size of 640 time frames by 128 Mel bands, without any vertical additional padding as can be seen in Figure 23b.

The output of the YOLOv5-based model consists of three resolution levels with spatial grid sizes of 20×20, 40×40, and 80×80, corresponding to different temporal and frequency scales in the input spectrogram. Each grid cell predicts a 14-dimensional vector encoding information about detected sound events.

To simplify the explanation, Figure 24 shows an illustrative example with smaller output grids of 5×5 and 10×10. Each cell in these grids predicts a 14-dimensional vector that encodes the properties of a potential sound event. The first value represents the activity probability, indicating the presence of an event; in the example, the values are 0.9, 0.7, and 0.8 for PRED 1, PRED 2, and PRED 3, respectively. The next two values correspond to the center of the event within the cell for the time and frequency axes, normalized between 0 and 1. For instance, the predicted positions are (0.2, 0.5) for PRED 1, and (0.3, 0.0) and (0.5, 0.0) for PRED 2 and PRED 3, respectively. Since the second grid has an

even number of frequency bins in this example, the event center aligns with the upper edge of the cell, resulting in a normalized frequency position of 0.0. The following two values represent the event's duration along the time axis and its extent along the frequency axis, both relative to the cell dimensions. In PRED 1, the temporal duration is smaller than the cell width, yielding a value of 0.8; in PRED 2, the event is longer than the cell, resulting in a predicted duration of 1.4; and in PRED 3, the duration equals the cell width, so the predicted value is 1.0. For the frequency span, the values are 5 for PRED 1—indicating the event spans 5 vertical cells—and 10 for both PRED 2 and PRED 3, meaning the event covers the full frequency range in the 10×10 grid. Finally, the remaining nine values encode the predicted class using a one-hot representation. All other cells in the grid produce zero-filled vectors, indicating no detected events.

It is important to note that PRED 1 and PRED 2 refer to the same underlying event but detected at different resolution levels. The model processes these predictions independently during training and inference.

To consolidate these multi-resolution detections and prevent redundant predictions, a post-processing step is applied. When the same event is detected in multiple cells—often across different resolutions—the predictions are merged by selecting the one with the highest confidence score as the predicted reference event. Other overlapping detections are discarded if they share a sufficient portion of area with the predicted reference event, based on intersection criteria (as presented in Equation 18). This strategy ensures that repeated predictions of the same event are suppressed, resulting in a more coherent and compact final output.

To effectively train the model, the output predictions are supervised using a composite loss function that combines classification, detection, and localization components. The classification aspect is handled using $\mathcal{L}_{BCE}$ (see Section 2.2.4), comparing the predicted one-hot encoded class distribution in each cell with the ground truth label. This component is weighted with a factor of 1.5.

Similarly, the detection component—often referred to as objectness loss—is also computed using $\mathcal{L}_{BCE}$ and determines whether an event is present in a particular cell. This component helps the model distinguish between active and inactive regions in the spectrogram and is assigned a weight of 0.5.

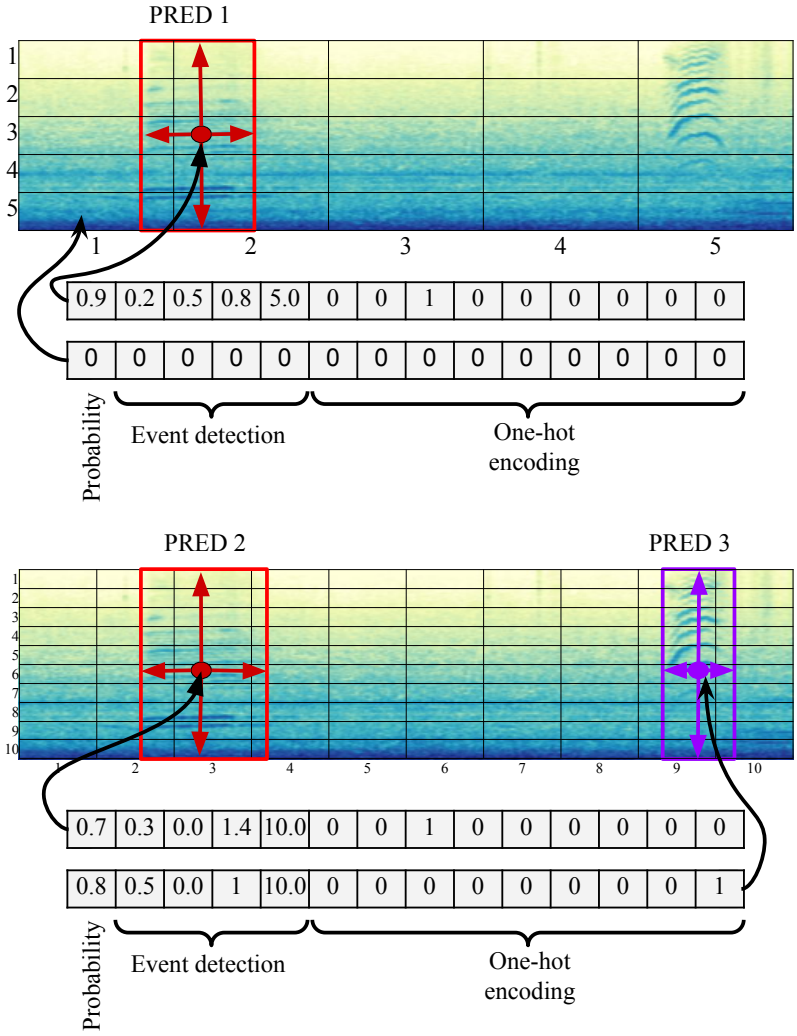


Figure 24: Illustration of event detection across multiple levels of resolution.

Finally, the localization component measures how accurately the model predicts the temporal position and duration of each event. Rather than using a simple distance metric like MSE, this loss is based on the $\mathcal{L}_{CIoU}$ (see Equation 20) between the predicted segment and the true label, promoting more precise temporal alignment [68]. This term is weighted by 7.5 to balance its influence during training.

4.2 Methodology

This section outlines the methodology used to evaluate the models in this study, based on the `SED_SYN` and `SED_REAL` datasets and their respective subsets, as explained in Section 3.3. The section provides details on the experimental setup, hardware configurations, and the evaluation metrics used to assess model performance.

4.2.1 Experiment Configuration

All models in this study were trained on a high-performance workstation equipped with an NVIDIA GeForce RTX 3060 GPU. This setup enabled efficient training by leveraging the GPU's parallel processing capabilities to accelerate convergence. Once training was complete, the models were exported to the ONNX format. This format offers a standardized model representation, facilitating interoperability across different deep learning frameworks and enabling deployment on a wide range of hardware platforms.

Inference was performed on four devices, summarized in Table 4. The first three—Jetson Nano, Raspberry Pi 4, and UDOO X86 II Ultra—represent edge computing environments characterized by limited computational power and energy constraints. These platforms were selected to assess model performance under real-world, resource-constrained conditions. The fourth device, a high-performance PC that only uses the CPU, was used as a baseline to illustrate the upper bound of the inference capabilities.

To optimize inference on edge devices, we explored model compression strategies (see Section 2.3.4). In particular, quantization was employed to reduce model size and improve execution speed, with minimal impact on accuracy. Although pruning was also initially considered, preliminary experiments revealed only marginal improvements in size and latency, and it was therefore excluded from the final implementation.

Table 4: Summary of hardware devices and their specifications used for model implementation and execution.

Name Device	Name CPU	Num. Cores	Freq. (GHz)	RAM (GB)	Power (W)
Jetson Nano	Arm Cortex-A57	4	1.43	4	10
Raspberry Pi 4	Arm V8 Cortex-A72	4	1.80	4	15
UDOO X86 II Ultra	Intel Pentium N3710	4	2.56	8	36
PC	Intel Core i7 9700k	8	3.60	16	900

4.2.2 Evaluation Setup

To assess model performance, we adopted both event-based and segment-based evaluation strategies. For event-based evaluation, we used the PSDS (see section 2.2.5), a standardized metric in the DCASE community, implemented via the `psds_eval` toolbox. PSDS is computed by constructing receiver operating characteristic (ROC) curves over 50 operating points, which allows the evaluation of detection quality across a wide range of thresholds.

The PSDS metric takes several factors into account, including the Detection Tolerance Criterion (DTC) and the Ground Truth Intersection Criterion (GTC), which define the conditions under which a detection is considered a true positive. In our evaluation, we employed two distinct PSDS variants: PSDS1, configured with $DTC = GTC = 0.7$, corresponds to the standard setup used in the DCASE challenge. This configuration enforces a stricter matching policy by requiring over 70% temporal overlap between predicted and reference events, thereby strongly penalizing inaccuracies in time localization. In contrast, PSDS0, with $DTC = GTC = 0.5$, is a proposed configuration that introduces more leniency, allowing for moderate temporal deviations in event boundaries. This setup better accommodates the variability inherent to our dataset. The use of both configurations enables a more nuanced assessment of model performance. Additionally, the best F1 score was calculated based on the detected events, reported as F1(1) and F1(0), where the number in parentheses corresponds to the respective PSDS variant used.

For segment-based evaluation, the model predictions were compared at fixed

time intervals of 0.1 seconds, treating each segment as a binary classification problem (see Section 2.2.5). Performance was measured using common metrics such as AUC, F1, ACC, and G-mean. These metrics were computed as micro-averages, meaning that TP, FP, and FN were aggregated across all classes before calculating the metrics, effectively treating all classes jointly as a single binary classification task. Additionally, the area under the ROC curve (AUC) was computed for all thresholds (θ), alongside a constrained AUC metric that limits false positive rates to below 0.1 ($AUC_{0.1}$), simulating practical deployment conditions.

To evaluate computational performance, a system was implemented to process 30 minutes of recorded audio, divided into 180 individual 10-second segments. In this setup, inference is not performed continuously; instead, it simulates real-time conditions, where new audio segments are recorded while predictions for the previous segment are processed concurrently. This approach replicates real-world scenarios in which recording and prediction occur simultaneously. The evaluation focused on key processes such as mel spectrogram computation, pre-processing, model inference, and post-processing, while also monitoring system resource utilization, including CPU and RAM usage, as well as processor temperature.

4.3 Results and Discussion

This section presents the experimental results obtained using the datasets described in Section 3.3. The discussion is structured in two main parts: detection performance and computational performance. The first part focuses on the model's ability to accurately detect and classify sound events across various evaluation metrics, providing insights into its effectiveness under real-world conditions. The second part examines computational aspects, including inference time, resource consumption, and system efficiency. This organization enables a clear distinction between the evaluation of detection accuracy and the assessment of practical deployment on edge devices.

Table 5: Performance comparison of CRNN and YOLOv5-based models on `SED_SYN_TEST` subset.

	CRNN	YOLOv5n	YOLOv5s	YOLOv5m
PSDS1	0.24	0.46	0.46	**0.47**
PSDS0	0.37	0.56	0.57	**0.59**
F1(1)	0.38	**0.60**	0.59	**0.60**
F1(0)	0.50	0.68	0.67	**0.69**
AUC	**0.92**	0.85	0.81	0.81
$AUC_{0.1}$	0.68	**0.78**	0.77	0.77
F1	0.65	0.75	**0.76**	**0.76**
ACC	0.95	0.96	0.96	**0.97**
G-mean	0.80	**0.84**	**0.84**	0.83

4.3.1 Detection Performance using a Synthetic Dataset

Table 5 summarizes the detection performance of the evaluated models on the `SED_SYN_TEST` subset. Across nearly all metrics, the CRNN model consistently underperforms when compared to its YOLO-based counterparts. This is especially evident in the event-based metrics: both PSDS1 and PSDS0 values are considerably higher for all YOLO variants. The difference is particularly notable under the relaxed configuration of PSDS0, which favors Recall, suggesting that CRNN struggles to detect complete events or accurately localize them. A similar trend can be observed in the F1(1) and F1(0) scores. These scores are computed using the optimal threshold selected per model, meaning the threshold is not fixed across models but adjusted individually to reflect each model's best possible F1 under the corresponding PSDS criterion.

Segment-based metrics (F1, ACC, G-mean) reinforce these findings. YOLOv5s and YOLOv5m achieve the best F1, pointing to strong segment-level performance. While ACC remains high across all models, the G-mean shows a modest but consistent advantage for YOLO models. It is important to note that these metrics are computed using the optimal threshold (θ) selected for each model based on its best F1, ensuring a fair and representative evaluation of performance under ideal thresholding conditions.

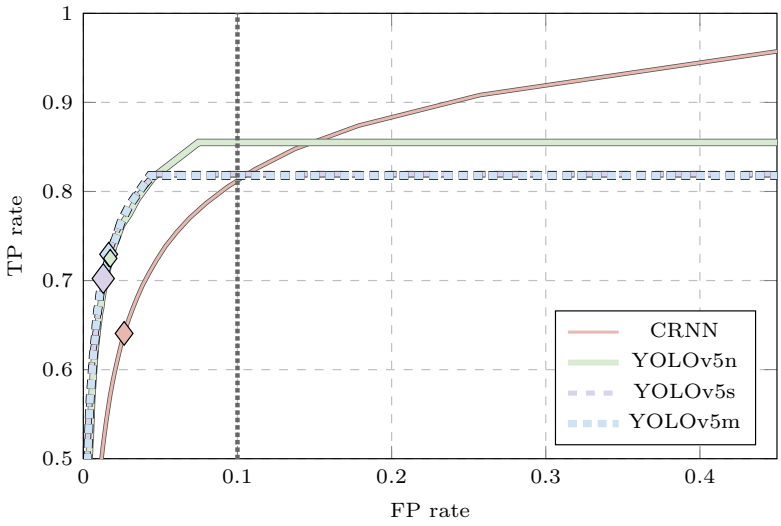


Figure 25: ROC curves on the subset `SED_SYN_TEST`.

Interestingly, CRNN achieves the highest AUC score, indicating that it ranks predictions effectively across thresholds. However, when considering the constrained AUC ($AUC_{0.1}$)—which only accounts for the operating region with false positive rate (FPR) below 0.1—the YOLO models clearly outperform. This highlights a key limitation of CRNN in more practical, deployment-like settings.

This behavior is better illustrated in Figure 25, which shows the ROC curves for all models. While CRNN reaches a true positive rate (TPR) close to 1 with low θ, YOLO-based models exhibit a form of saturation where their TPR plateaus before reaching maximum values. This effect arises from the YOLO models' tendency to output exact zero probabilities for certain classes in some cells, effectively discarding those detections regardless of the threshold. As a result, even the lowest evaluated θ values fail to recover these missed events. Although this limits the models' maximum TPR and slightly reduces their overall AUC, their performance in constrained operating regions—where false positives must be minimized—is more consistent and ultimately superior, as reflected in their higher $AUC_{0.1}$ scores.

Another way to analyze the stability of the models across different threshold

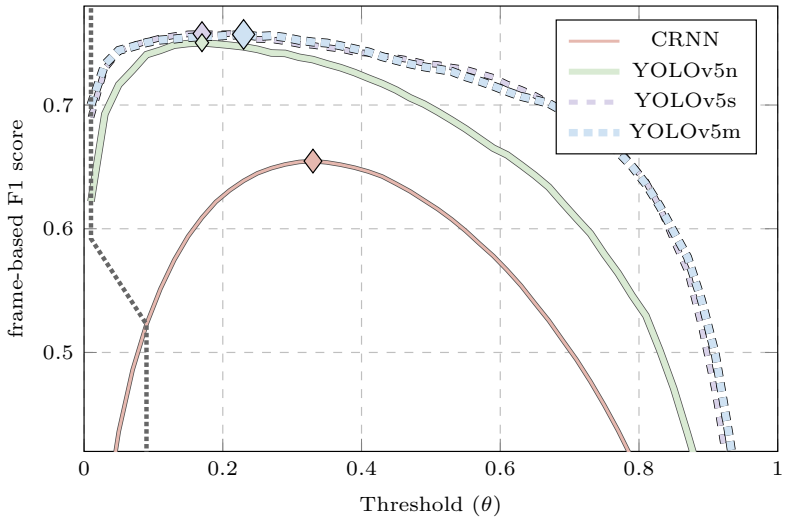
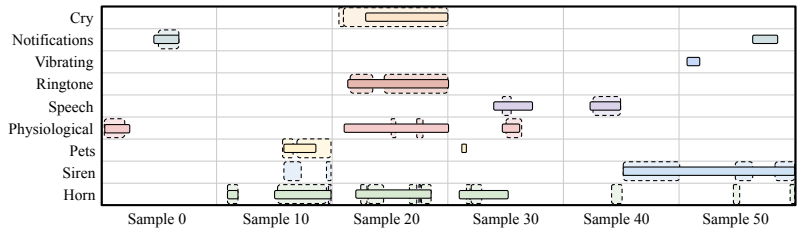


Figure 26: F1 evaluation across various threshold values for `SED_SYN_TEST`.

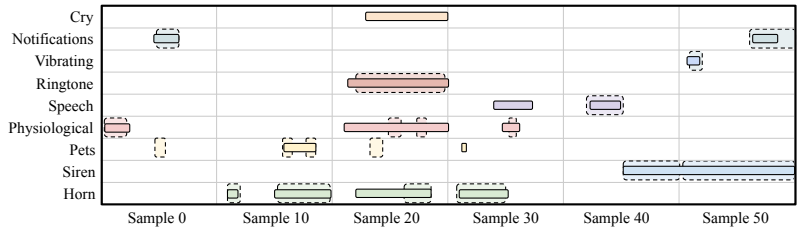
values is by using a plot of F1 vs θ. Figure 26 presents the performance of the models across a range of thresholds, illustrating how the F1 evolves as the threshold changes. In this graph, it is evident that YOLO-based models exhibit a more stable F1, indicating a lower dependence on the θ parameter. This suggests that these models maintain consistent performance across various operational conditions. In contrast, the CRNN model shows a more pronounced parabolic-like trend compared to the YOLO-based models, with performance peaking at a specific threshold value. This behavior indicates that the CRNN model is more sensitive to the selection of θ, achieving optimal performance only within a comparatively narrower range of threshold values.

In both Figure 25 and Figure 26, the diamond markers indicate the threshold values at which each model achieves its maximum F1. These optimal points correspond to the values reported in Table 5.

The more gradual variation observed in the YOLO-based models' F1 curves highlights a key advantage: their robustness and reliability across different threshold values. This stability makes them particularly suitable for real-world applications, where precise threshold tuning is often impractical or subject to



(a) CRNN (θ=0.33)



(b) YOLOv5n (θ=0.17)

Figure 27: Prediction outputs filtered at given thresholds θ for six particular audio segments from the `SED_SYN_TEST`.

change over time.

Finally, a graphical representation of the predictions for a selection of samples using the optimal θ values is presented in Figure 26 for both the CRNN and YOLOv5n models. The true labels are represented by solid lines, while the predictions are shown as dashed lines. Overall, we can notice that the behavior of both models is quite similar. However, two issues stand out in the CRNN predictions: first, there is a higher number of false positives, particularly in the Siren and Horn classes; second, some predictions appear fragmented, failing to cover the entire event, as seen in sample 20 (class Ringtone) and sample 50 (class Siren), which are heavily penalized.

These observations, based on the synthetic dataset, collectively suggest that YOLO-based models consistently outperform the CRNN baseline across all evaluation metrics and visual analyses. This advantage is particularly evident in event-based metrics, where YOLO models demonstrate better detection coverage and temporal localization of events, as well as more consistent segment-level performance. The graphical representations further highlight fewer false positives

and more complete detections compared to CRNN. Notably, the performance differences among the YOLOv5 variants (n, s, m) remain relatively small, despite their increasing parameter counts—approximately 1.9M, 7.2M, and 20M, respectively. This suggests that increasing model complexity results in comparable classification performance, which may be attributed to the relatively low number of event classes (9), where the feature representations learned by the lightweight YOLOv5n model appear sufficient to capture the relevant audio characteristics.

4.3.2 Detection Performance using Real Datasets

Similarly to the synthetic scenario, we conducted an analysis on the `SED_REAL` dataset, starting with the `SED_REAL_S1` subset. This real-world scenario presents inherent limitations, particularly regarding the number of annotated events, as discussed in Section 3.3. These constraints must be considered when interpreting the models' performance.

When comparing models under these real-world conditions, YOLO-based architectures still appear to outperform the CRNN baseline. However, the performance gap between the two approaches has narrowed considerably. Both model families experience a substantial drop in their scores across most metrics, a trend primarily driven by the limitations of the `SED_REAL_S1` subset—namely, the reduced number of annotated events and their uneven distribution. These constraints not only affect detection metrics but also reduce the reliability of threshold selection and increase sensitivity to false positives. Nevertheless, the overall patterns observed suggest that YOLO-based models maintain a robustness advantage even under more challenging and imperfect real-world conditions. A summary of the detection performance for this evaluation is presented in Table 6.

For event-based metrics such as PSDS1 and PSDS0, all models achieve relatively low scores. Among the YOLO variants, YOLOv5m obtains the highest PSDS0 value (0.10), indicating a slight advantage in event recall under relaxed matching criteria. In contrast, CRNN performs worse across both PSDS metrics, particularly PSDS1, suggesting reduced effectiveness in capturing

Table 6: Detection performance on the SED_REAL_S1 subset.

	CRNN	YOLOv5n	YOLOv5s	YOLOv5m
PSDS1	0.01	0.01	0.03	**0.04**
PSDS0	0.04	0.01	0.03	**0.10**
F1(1)	0.18	0.40	**0.44**	0.39
F1(0)	0.25	**0.52**	0.44	0.39
AUC	0.87	**0.92**	0.89	0.87
$AUC_{0.1}$	0.51	0.71	**0.78**	0.73
F1	0.59	0.71	**0.77**	0.73
ACC	0.93	0.94	**0.95**	0.94
G-mean	0.79	0.89	**0.90**	0.89

complete events. The F1(1) and F1(0) scores—computed at the optimal θ for each model—further indicate that YOLO-based models, especially YOLOv5s and YOLOv5n, outperform CRNN. As shown in Figure 28, these optimal F1(1) are represented by upward-pointing triangles. Downward-pointing triangles denote the F1(1) obtained when applying the best threshold determined from the synthetic partition. A noticeable gap can be observed between these two settings for all models, highlighting the sensitivity of threshold selection in domain transfer. Nevertheless, even when using the suboptimal configuration where the threshold is selected based on the synthetic dataset, YOLO-based models still outperform the CRNN baseline, reaffirming their robustness and generalization capacity.

In terms of segment-based metrics, YOLOv5n achieves the highest AUC (0.92), while YOLOv5s obtains the highest constrained AUC ($AUC_{0.1}$) with a score of 0.78. Although CRNN reaches a moderate AUC of 0.87, its significantly lower $AUC_{0.1}$ (0.51) underscores its reduced reliability under stricter conditions.

Finally, the general performance metrics—F1, ACC, and G-mean—follow a consistent pattern. YOLOv5s again leads in accuracy (0.95) and geometric mean (0.90), while CRNN shows the lowest performance across all three metrics. It is important to note that these results are reported using the best threshold θ for each model, selected to maximize the F1 score on the corresponding evaluation partition.

We also evaluated the models on the SED_REAL_S2 subset, which, unlike

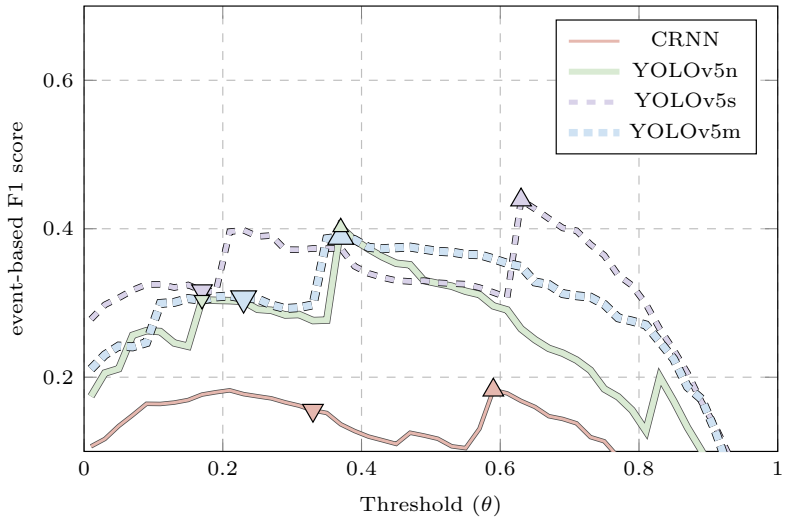


Figure 28: F1(1) values across all thresholds for each model on the `SED_REAL_S1` subset.

`SED_REAL_S1`, features a more representative distribution of events. This makes it particularly valuable for assessing model behavior under conditions closer to real-world deployments. Given the performance similarities previously observed among the YOLOv5 variants (n, s, and m), and considering our focus on lightweight solutions, we restrict this analysis to CRNN and YOLOv5n. This selection emphasizes the trade-off between model complexity and performance, aligning with the goal of identifying efficient models for practical use.

Table 7 presents the detection results for `SED_REAL_S2`. Overall, YOLOv5n maintains a consistent advantage over CRNN across most metrics, particularly in event-based measures such as PSDS0, where it more than doubles the CRNN score, even though both models achieve similar F1(0) values. A similar pattern is observed for PSDS1: despite comparable F1(1) scores, YOLOv5n achieves noticeably higher performance. While these PSDS values remain relatively low in absolute terms, they represent a clear improvement over those observed in `SED_REAL_S1`, highlighting that the improved performance is not only model-dependent, but also reflects the more favorable characteristics of

Table 7: Detection performance on the SED_REAL_S2 subset.

	CRNN	YOLOv5n
PSDS1	0.00	**0.06**
PSDS0	0.14	**0.36**
F1(1)	0.23	**0.25**
F1(0)	0.29	**0.30**
AUC	**0.96**	0.95
$AUC_{0.1}$	0.80	**0.84**
F1	0.77	**0.81**
ACC	0.96	**0.97**
G-mean	0.85	**0.90**

SED_REAL_S2, including a richer and more representative event distribution.

Segment-based metrics show more comparable results between the models. Still, YOLOv5n performs slightly better in terms of constrained AUC ($AUC_{0.1}$), ACC, and F1. Notably, these values represent an improvement over those observed in SED_REAL_S1, reflecting the enhanced event representation and data quality in SED_REAL_S2.

Performance on the SED_REAL_S2 subset is generally higher than that observed for SED_REAL_S1, with improvements in PSDS1, PSDS0, $AUC_{0.1}$, and ACC. However, a slight decrease is noted in both F1(1) and F1(0). These differences can be primarily attributed to the number of event classes considered: while SED_REAL_S1 includes only five classes with annotated events, SED_REAL_S2 contains eight, potentially increasing the number of false positives and affecting the F1 scores.

An additional observation concerns the threshold values used for detection. As shown in Figure 29, the thresholds (θ) that maximize the F1 on the SED_SYN_TEST (represented by downward-pointing triangles) are now closer to the optimal thresholds computed for the SED_REAL_S2 subset (upward-pointing triangles). This improved alignment between domains suggests greater robustness in threshold selection when more representative and diverse real data are available.

Although some evaluation metrics suggest suboptimal performance of the models under real-world conditions, Figure 30 presents a more nuanced view. It displays six consecutive audio segments from the SED_REAL_S1 subset, comparing

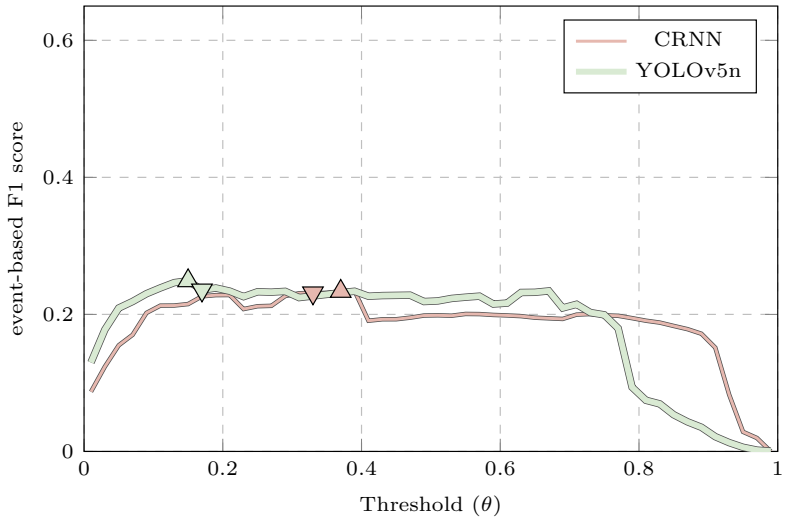
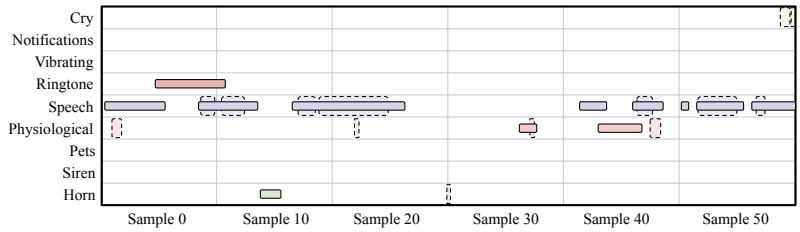


Figure 29: F1(1) values across all thresholds for each model on the `SED_REAL_S2` subset.

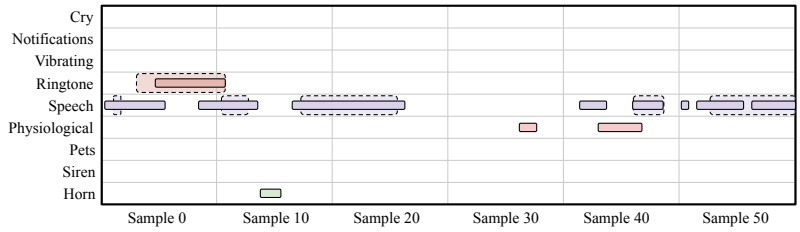
the labels (solid lines) with the model predictions (shown in dashed lines). Despite low scores in event-based metrics like PSDS, both CRNN and YOLOv5n correctly identify instances of the Ringtone and Speech classes. This highlights a potential mismatch between metric outcomes and actual perceptual performance. In this particular scenario, event-based metrics may be overly stringent—especially when events are long and predictions fail to fully cover their duration. Additionally, given the reduced number of annotated events, FP weigh more heavily in the evaluation, further penalizing the metrics. In contrast, frame-based evaluations appear to better reflect the models' effective behavior in these cases.

4.3.3 Impact of Quantization on Detection Performance

This section analyzes the impact of model quantization on detection performance for both YOLO and CRNN architectures. In contrast to previous sections that focused separately on synthetic and real-world scenarios, the current evaluation



(a) CRNN (θ=0.59)



(b) YOLOv5n (θ=0.37)

Figure 30: Prediction outputs filtered at given thresholds θ for six in a row audio segments from the `SED_REAL_S1` subset.

encompasses both domains to examine how compression techniques affect the trade-off between model size and accuracy. For the real-world evaluation, the `SED_REAL_S2` subset is used—a more complete subset of the `SED_REAL_S1`.

The remainder of the analysis considers only CRNN and YOLOv5n models, selected due to its favorable trade-off between detection performance and model size, as observed in previous sections. Table 8 summarizes the detection performance of YOLOv5n and CRNN under three quantization configurations—no quantization, dynamic quantization, and static quantization—evaluated on the `SED_SYN_TEST` and `SED_REAL_S2` subsets. To maintain clarity while capturing both temporal and event-level aspects of detection, we focus on two representative metrics: F1 and PSDS(0). The F1 reflects frame-based performance, indicating how well predictions align with ground truth over time, while PSDS(0) provides a less restrictive event-based evaluation, better suited to the nature of sound event data.

On the synthetic dataset, YOLOv5n shows a relatively minor performance drop under quantization. Its F1 score decreases marginally from 0.75 to 0.74,

Table 8: Detection performance of YOLOv5n and CRNN models with and without static and dynamic quantization.

		Synthetic		Real	
		F1	PSDS(0)	F1	PSDS(0)
YOLOv5n	No Q.	0.75	0.56	0.81	0.36
	Dynamic Q.	0.74	0.50	0.80	0.35
	Static Q.	0.74	0.48	0.80	0.32
CRNN	No Q.	0.65	0.37	0.77	0.14
	Dynamic Q.	0.58	0.10	0.76	0.15
	Static Q.	0.63	0.13	0.76	0.04

while PSDS(0) drops more noticeably—from 0.56 to 0.48 in the static quantized version. This suggests that YOLOv5n maintains robust detection capabilities even under compression, although certain aspects of temporal alignment (captured by PSDS(0)) may be more sensitive to quantization noise.

In contrast, CRNN exhibits a sharper degradation in performance when quantized. The static quantization case is particularly affected, with PSDS(0) plummeting from 0.37 to 0.13 on the synthetic dataset, while F1 remains somewhat stable. This discrepancy highlights how the sequential and recurrent nature of CRNN may be more vulnerable to the reduced numerical precision introduced by quantization—especially for metrics that rely on temporal consistency.

When evaluating on the `SED_REAL_S2` subset, similar trends are observed. YOLOv5n continues to outperform CRNN across all quantization modes. Although all models show slightly reduced PSDS(0) scores under static quantization, the drop is less severe for YOLOv5n. Interestingly, dynamic quantization appears to be the more favorable technique for CRNN on the real dataset, yielding a higher PSDS(0) score (0.15) compared to static quantization (0.04). However, even in the best case, CRNN does not reach the performance levels of the YOLO model.

This improvement in PSDS(0) scores—from 0.01 to 0.36 for YOLOv5n, and from 0.04 to 0.14 for CRNN—when moving from the `SED_REAL_S1` to the more comprehensive `SED_REAL_S2` subset highlights the sensitivity of event-based metrics to dataset composition. The increased number and diversity of annotated

events in SED_REAL_S2 provide a more stable evaluation setting, reducing the disproportionate influence of isolated detection errors and better reflecting the models' true capabilities in real-world scenarios.

Overall, these results suggest that YOLOv5n not only offers superior performance in uncompressed form, but also maintains higher robustness under both static and dynamic quantization. This robustness makes it a strong candidate for real-time or embedded sound event detection systems, where model size and inference efficiency are critical.

4.3.4 Computational Performance

To complement the previous analysis on detection, this section evaluates the computational performance of the YOLOv5n and CRNN models across different hardware platforms. Figure 31 presents the inference times for each model—unquantized, dynamically quantized, and statically quantized—measured across 180 runs on four distinct CPU platforms: Intel N3710, Cortex-A57, Cortex-A72, and Intel Core i7.

In all tested configurations, YOLOv5n consistently outperforms CRNN in terms of inference speed. The boxplots in Figure 31 show significantly lower median and dispersion values for YOLOv5n, indicating both faster and more stable performance across platforms.

Quantization has varying effects depending on the model and platform. For YOLOv5n, the impact of both static and dynamic quantization on inference time is minimal, suggesting that the architecture is already well-optimized for efficient execution. In contrast, the CRNN model shows a notable reduction in inference time when statically quantized, particularly on edge devices such as the Cortex-A57 and Cortex-A72. However, this speed-up is less evident on high-performance CPUs like the Intel Core i7, where other factors such as memory access and threading may dominate execution time.

From a practical standpoint, the Cortex-A72 platform—commonly found in affordable and widely available Raspberry Pi devices—offers a viable deployment target. On this processor, the YOLOv5n model processes a 10-second audio

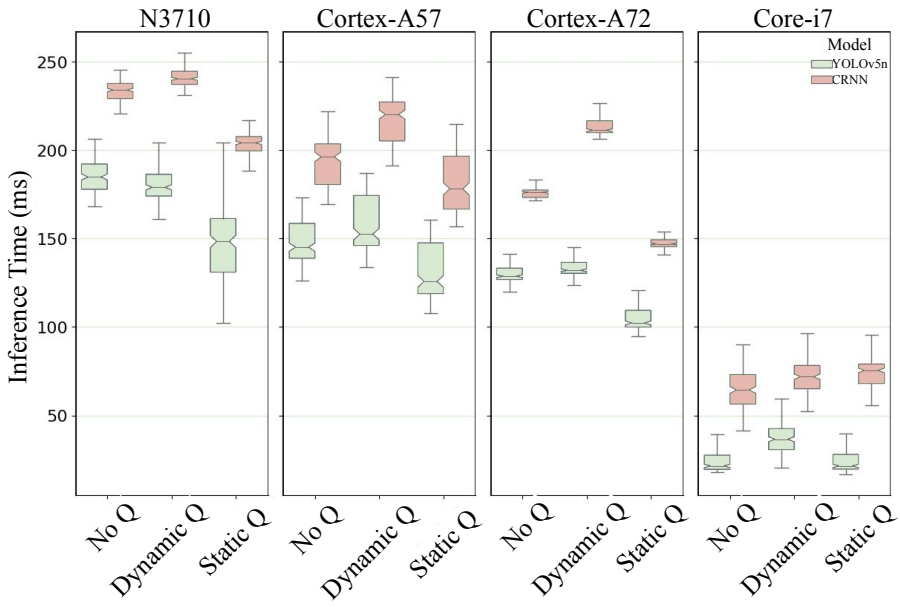


Figure 31: Inference time distributions (in milliseconds) for the CRNN and YOLOv5n models.

segment in approximately 100–150 milliseconds. This level of performance allows for real-time or near-real-time detection, and even enables overlapping or repeated inference over the same audio window to increase robustness.

In addition to inference, it is crucial to consider other stages in the full execution pipeline, such as feature extraction, preprocessing, and postprocessing. Figure 32 compares the time required for each of these stages for both the CRNN and YOLOv5n models. As shown, feature extraction takes a similar amount of time for both models, as they both rely on the same Mel spectrogram input. However, preprocessing is slightly more costly for YOLOv5, as it requires padding to match the expected input shape, whereas CRNN can process the original spectrogram directly. Similarly, postprocessing tends to be more time-consuming for YOLOv5, due to the additional steps needed to filter detections referring to the same event. Despite these differences, both preprocessing and postprocessing are negligible compared to inference and feature extraction, as each takes less than 3 ms to complete.

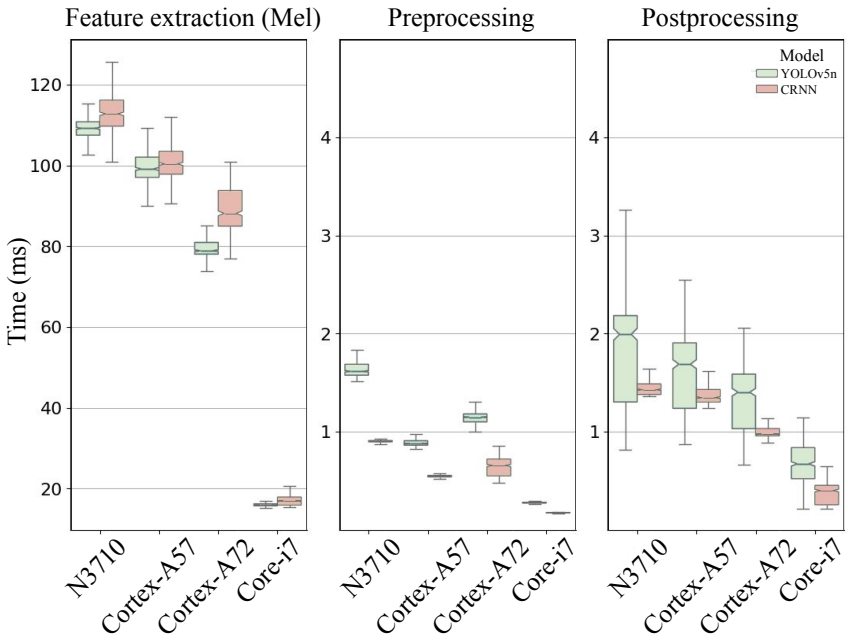


Figure 32: Time for feature extraction, preprocessing, and postprocessing stages across CRNN and YOLOv5n models.

In Table 9, we compare memory usage, operating temperature, and model size for both the CRNN and YOLOv5n models across different hardware platforms.

Regarding temperature, we observe that both models maintain relatively stable temperatures across all platforms, with no significant differences between them. However, it is worth noting that the Cortex-A57 processor, being the least powerful in terms of raw performance, shows the lowest temperatures for both models, with a stable temperature of around 32°C.

For RAM usage, the results are similar for both models, as they both utilize almost the same amount of memory during processing. There are no significant discrepancies between them in this aspect, meaning that both models have similar memory requirements regardless of the platform or quantization technique used.

Finally, when it comes to model size, there is a notable difference. YOLOv5n is almost twice as large as CRNN, with a model size of 9.63 MB compared to 4.25 MB for CRNN. Despite this larger size, YOLOv5n still outperforms CRNN

Table 9: Mean values and standard deviations where appropriate, corresponding to memory (RAM) usage and operating temperature when using CRNN and YOLOv5n models.

			No Q.	Dynamic Q.	Static Q.
Temperature (ºC)	YOLOv5n	N3710	58.44 ± 1.35	62.00 ± 1.26	56.53 ± 1.49
		Cortex-A57	32.09 ± 0.54	32.04 ± 0.48	31.26 ± 0.50
		Cortex-A72	51.25 ± 0.51	50.94 ± 0.54	50.45 ± 0.55
		Core-i7	51.53 ± 0.67	44.00 ± 0.76	51.25 ± 1.35
	CRNN	N3710	52.18 ± 0.54	51.46 ± 1.02	52.56 ± 0.55
		Cortex-A57	32.29 ± 0.69	30.93 ± 0.74	30.59 ± 0.86
		Cortex-A72	51.25 ± 0.51	50.94 ± 0.54	50.45 ± 0.55
		Core-i7	53.23 ± 0.84	52.36 ± 1.21	52.51 ± 0.72
RAM (MB)	YOLOv5n	N3710	225.56 ± 0.48	221.10 ± 0.72	223.06 ± 0.46
		Cortex-A57	229.15 ± 1.49	223.39 ± 0.97	227.94 ± 0.82
		Cortex-A72	244.14 ± 0.85	240.07 ± 0.72	239.39 ± 0.71
		Core-i7	250.08 ± 0.46	243.95 ± 0.70	248.25 ± 0.71
	CRNN	N3710	257.06 ± 2.34	260.61 ± 2.34	229.88 ± 0.78
		Cortex-A57	262.74 ± 2.55	264.40 ± 2.35	230.66 ± 0.71
		Cortex-A72	278.95 ± 2.36	276.87 ± 2.34	243.65 ± 1.48
		Core-i7	286.15 ± 2.33	286.64 ± 2.37	253.88 ± 0.38
Model size (MB)	YOLOv5n	All	9.63	2.60	2.60
	CRNN	All	4.25	2.52	2.52

in terms of inference time. After applying quantization techniques, both models reduce significantly in size and approach a similar model size, making them more comparable in terms of storage requirements and deployability.

4.4 Strengths and Limitations of the YOLO-Based Approach

The experimental results presented throughout this work highlight several strengths of the YOLO-based approach for sound event detection (SED). Most

notably, the YOLOv5n variant achieves a strong trade-off between detection accuracy and model size, consistently outperforming the CRNN baseline across synthetic and real-world conditions. Its lightweight architecture—combined with efficient inference capabilities—makes it especially suitable for real-time or resource-constrained deployment scenarios.

Another key advantage lies in the robustness of YOLO models under quantization. Results indicate that detection performance remains stable under both dynamic and static quantization, particularly for YOLOv5n, suggesting that compression can be applied without significant degradation. Additionally, segment-based metrics (e.g., AUC, $\text{AUC}_{0.1}$) and visual analyses consistently reflect more complete and accurate event representations for YOLO models, supporting their effectiveness beyond purely numerical scores.

However, several limitations should be considered. First, the evaluation reveals that event-based metrics, such as PSDS, are highly susceptible to the composition of the evaluation dataset. For instance, PSDS(0) values increased notably from 0.01 (YOLOv5n on `SED_REAL_S1`) to 0.36 (on `SED_REAL_S2`) when moving to a more comprehensive and balanced dataset, exposing the dependency of certain metrics on the number and distribution of annotated events.

Second, while synthetic data allows for systematic benchmarking, it does not fully represent the variability and acoustic complexity of real-world environments. Generalizing these findings to more challenging or diverse settings may therefore require further validation.

In addition, the proposed YOLO-based architecture remains closely tied to its original formulation for image processing. While its adaptation to the audio domain yields strong results, further improvements may be possible by incorporating audio-specific architectural elements, such as enhanced temporal modeling or frequency-aware processing.

It is important to clarify that the CRNN model evaluated here does not represent the most advanced configurations available in the literature. More sophisticated variants—especially those utilizing weakly labeled data or large pretrained feature extractors—were intentionally excluded to ensure a fair and controlled comparison with our proposed YOLO-based approach, which

currently does not support weak supervision nor relies on extensive pretrained backbones. Nonetheless, these advanced CRNN models have demonstrated strong performance in prior studies and remain relevant benchmarks. Bridging the gap between weakly supervised learning and YOLO architectures could enable leveraging larger, less curated datasets and further improve generalization. Likewise, integrating pretrained audio embeddings into the YOLO framework could enhance performance while preserving inference efficiency.

Chapter 5:
Incremental Domain Adaptation Strategy for Sound Events

In the previous chapter, we focused on the detection of particularly distracting sound events in driving environments. As discussed in Section 3.2, a key factor that contributes to distraction is the emotional content of interactions, especially those occurring between the driver and passengers. It is not the same when a driver engages in a conversation while angry as when in a neutral emotional state; emotions can be decisive in estimating the level of distraction in certain situations.

Similar to the detection of distracting events, the recognition of emotions—especially when relying solely on audio sources—represents a significant challenge due to the large variability involved. This effect can be even more pronounced than in the detection of events like sirens or horns, since domain shifts are not only caused by environmental factors such as background noise or microphone differences, but also by the fact that different individuals may express the same emotion in very different ways. Additionally, defining the boundaries of a particular emotion can be a complex and subjective task. The field of Speech Emotion Recognition (SER) is dedicated to studying the recognition of emotions conveyed through speech and addresses these inherent challenges.

This chapter introduces the third main contribution of this thesis: an incremental domain adaptation strategy for sound event classification. Although it is initially focused on the task of Speech Emotion Recognition (SER), the strategy is designed to be applicable to more general domain shift scenarios—for example, changes in background noise or environmental conditions in the context of distraction detection. In the case of SER, a domain shift typically occurs when a new speaker is introduced, which often results in a significant drop in recognition performance.

In the remainder of this chapter, the generalization problem across speakers in SER is examined, and the potential of incremental learning to mitigate this issue is explored. Our innovative proposal is explained in depth, highlighting its design and key components. The experimental setup and evaluation protocol used to assess these strategies are then described, including the structure employed to simulate incremental learning scenarios. Finally, results are presented and analyzed, comparing the performance of the different selection methods and evaluating their effectiveness in maintaining generalization under domain shift conditions.

5.1 Speech Emotion Recognition

Speech Emotion Recognition is a well-established task within the broader field of affective computing, concerned with automatically identifying the emotional state conveyed by a speaker through their voice. Recent advances in deep learning have significantly improved performance in SER by enabling models to capture complex relationships in acoustic data.

Among the most commonly used architectures for SER are CNNs [132, 133], RNNs [134, 135], and more recently, Transformers [15, 136]. CNNs are typically used for feature extraction from spectrograms, while RNNs, particularly LSTMs [137, 138] and GRUs [139] networks, have been effective for capturing temporal dependencies in speech. However, Transformers have gained significant attention in recent years due to their ability to model long-range dependencies and capture contextual information across entire sequences. This is especially relevant for SER, as accurately identifying an emotion in speech requires not only analyzing the prosody and tone but also partially understanding the message conveyed by the speaker [15, 136].

Although SER has predominantly been framed as a classification task throughout this thesis, it is worth noting that the problem can also be approached from a regression perspective. In this alternative formulation, emotions are represented using continuous values along two dimensions: Arousal and Valence, as proposed by the circumplex model of emotion [140]. Arousal reflects the

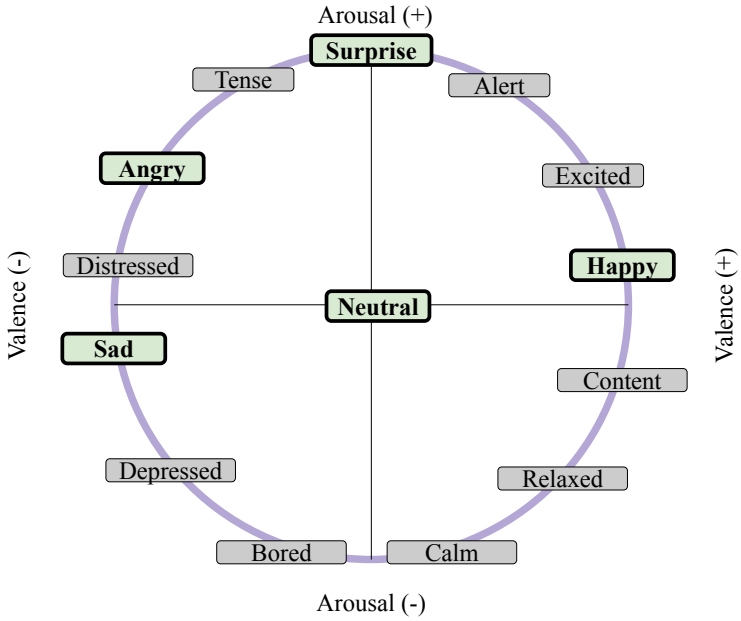


Figure 33: Arousal-Valence circumplex model illustrating the positioning of various emotions.

intensity or activation level of the emotion (from calm to excited), while Valence captures the emotional tone, ranging from positive (pleasant) to negative (unpleasant). Although this dimensional representation offers a more nuanced view of affective states, the present work focuses exclusively on categorical emotion labels, which are highlighted in green in Figure 33, whose layout is based on a commonly used version of the circumplex model[1].

While the discrete classification approach has clear advantages, such as simplifying the labeling process, it can be limited by the subjectivity involved in defining emotion boundaries. This makes it challenging to determine clear-cut boundaries between similar emotions. In contrast, the continuous labeling approach, while offering a more nuanced representation of emotions, introduces greater variability in the assigned positions, as different annotators may perceive and label the same emotional expression differently. These limitations underscore

[1] https://upload.wikimedia.org/wikipedia/commons/a/ad/Circumplex_model_of_emotion.svg

the trade-offs between the two approaches and the complexities inherent in SER systems.

SER has multiple real-world applications [141, 142, 143, 144], ranging from virtual assistants to enhancing user experiences in mobile devices, with critical areas such as road safety being especially relevant. This is because, as discussed earlier in the Section 3.2, a driver's emotional state can significantly influence how events are perceived and, consequently, the driver's behavior on the road. In emotionally charged situations, such as anger or sadness, the driver's ability to process information and respond appropriately can be severely impaired, increasing the risk of accidents.

5.1.1 Speaker-Independent SER

A common evaluation setting in SER is the speaker-independent scenario, in which the system is tested on data from speakers not seen during training. This setting is crucial for assessing a model's ability to generalize beyond the specific voices it was trained on and is more representative of real-world conditions.

However, one of the central challenges in speaker-independent SER lies in the variability introduced by different speakers. When SER models are trained and evaluated using data from the same set of speakers, they often achieve high performance due to the consistency in vocal characteristics and emotional expression styles. Yet, this performance tends to drop significantly when models are tested on unseen speakers [145].

This degradation is primarily attributed to the lack of generalization across speakers. Each individual has unique vocal traits, such as pitch range, speaking rate, accent, and prosody, which affect how emotions are expressed and perceived in speech. Consequently, models trained on a limited and homogeneous set of speakers may overfit to speaker-specific patterns rather than learning generalizable emotional cues.

To address the generalization limitations inherent in speaker-independent SER, various approaches based on domain adaptation have been proposed. These methods aim to reduce the mismatch between the training (source domain,

SD) and testing (target domain, TD) distributions by adapting the learned representations to better reflect the characteristics of unseen speakers. Some strategies focus on explicitly measuring and minimizing domain shifts [146], while others attempt to enhance model robustness through improved training objectives or architectural modifications that promote domain-invariant feature learning [147].

5.1.2 Speaker Adaptation for SER

While speaker-independent SER aims to build models that generalize across a wide range of unseen speakers, a related and more targeted line of research is speaker adaptation. Instead of optimizing for broad generalization, speaker adaptation techniques aim to adjust a pre-trained model to better perform on data from a specific new speaker—the target domain—by leveraging what has already been learned from the source domain. This approach often assumes limited labeled data from the new speaker but can also exploit larger amounts of unlabeled data, adapting the model incrementally or with minimal retraining, which is particularly useful in real-world scenarios where data availability is constrained and speaker variability is high [148, 149].

Figure 34 illustrates the three primary paradigms commonly found in SER: speaker-dependent, speaker-independent, and speaker adaptation. In the speaker-dependent setting, the same speakers are used for both training and testing, typically leading to high but potentially over-optimistic results. The speaker-independent setup, by contrast, evaluates the model on completely unseen speakers, providing a better estimate of generalization but often revealing a significant drop in performance. Speaker adaptation lies in between these two extremes: the model is first trained on a diverse set of source speakers and subsequently adapted to a single unseen target speaker, balancing generalization and personalization.

When speaker adaptation is performed using unsupervised techniques, a common methodology is to use the predicted labels from a base model and, based on a selection criterion, retain only those with the highest probability of being

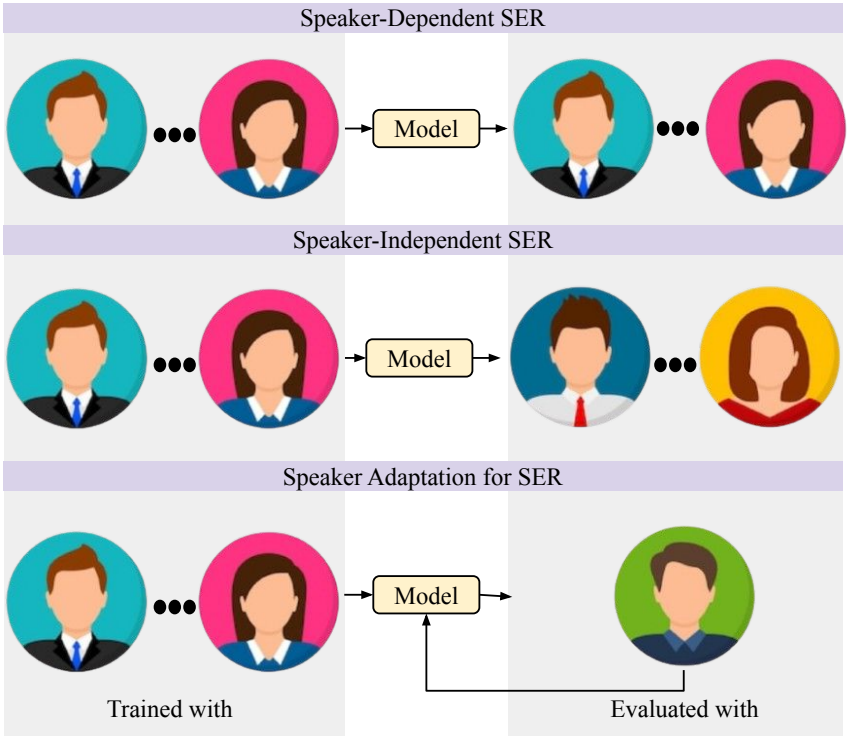


Figure 34: Overview of the three main paradigms in Speech Emotion Recognition.

correct. These newly labeled data are then used to adjust the model, and this process can be repeated iteratively [148]. However, this strategy has two main limitations: first, a strong dependence on the quality of the initial model, which must generate sufficiently accurate predictions; and second, the need to correctly discriminate which labels are reliable for retraining.

On the other hand, when the problem is approached from a supervised perspective, the most straightforward strategy is to fine-tune the model using correctly labeled samples, although this is limited by the typically small number of such data available [150]. This approach eliminates the problems associated with unsupervised strategies by relying on true labels but introduces new critical challenges, such as determining how many and which samples are sufficient to effectively adapt the model to the new speaker.

Our semi-supervised proposal builds on the limitations of both approaches:

first, we perform an unsupervised selection of samples that are potentially relevant for effective adaptation; then, we aim to minimize the number of these samples so that a user can manually label them, using this information to update the model. This proposed approach will be explained in detail in the next section.

5.1.3 Incremental Strategy for Speaker Adaptation

In applications such as in-vehicle emotion recognition, where the goal is to build a system capable of recognizing the emotions of a specific speaker—typically the driver—it is often impractical to collect a fully labeled dataset for that individual and train a dedicated model. Furthermore, because of the wide variability between speakers, relying on a single speaker-independent model that generalizes well to everyone is unrealistic. To overcome these challenges, we propose a strategy called Incremental Selection and Adaptation (ISA), which allows models to adapt to different speakers with minimal user feedback. In this approach, users are required to label only a small number of samples that the system incrementally selects and presents for annotation as new unlabeled data becomes available.

Our approach, illustrated in Figure 35, begins with an initial model, referred to as $Model_0$, which has been pre-trained on a source domain (SD) dataset with multiple speakers and labeled.

The representations generated by this model over the target domain (TD)—which corresponds to unlabeled data from a new speaker—are used by a sample selection method to extract a representative subset of TD. Once labeled, this subset is combined with samples from SD to form a small set of selected samples (SS), which is then used to retrain the model. In each incremental step, SS grows as new samples are selected, allowing for progressive and informed model adaptation.

The model architecture is composed of three main components designed to balance robustness and adaptability. At its core lies a frozen feature extractor based on Wav2Vec 2.0, which has been fine-tuned specifically for emotion recognition [15]. This module processes raw audio input (data in the original

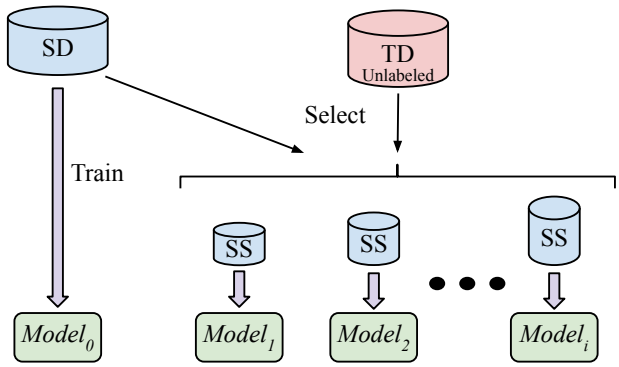


Figure 35: General overview of the proposed incremental semi-supervised speaker adaptation strategy.

representation space) and produces a high-dimensional embedding that captures the essential characteristics of the entire speech segment.

Building upon this, a domain adapter block—composed of fully connected layers—transforms and reduces the dimensionality of the embeddings into a more compact latent representation space. This space is essential for the selection method, which identifies key samples to personalize the model for a new speaker. Finally, the classification block maps these latent representations to categorical emotion predictions. Importantly, only the domain adapter and classification blocks are retrained during speaker adaptation, allowing the model to efficiently personalize while preserving stable and transferable core acoustic features. A schematic of this architecture is shown in Figure 36, where x denotes a single raw input sample (with $x \in X$), $\hat{y}$ its predicted categorical label, and $\hat{x}$ its corresponding latent representation, with dimensionality $\hat{x} \in \mathbb{R}^z$, where z is the number of neurons in the final layer of the domain adapter block.

The model is trained using a loss function combining $\mathcal{L}_C$ and $\mathcal{L}_{CCE}$ (see Sections 2.2.4 and 2.2.4). This loss function operates on the latent representation space and aims to cluster samples of the same class into compact regions, while $\mathcal{L}_{CCE}$ promotes separation between different classes. This structure in the latent representation space facilitates the selection method by making key samples more distinguishable. This training strategy is applied both during the initial training

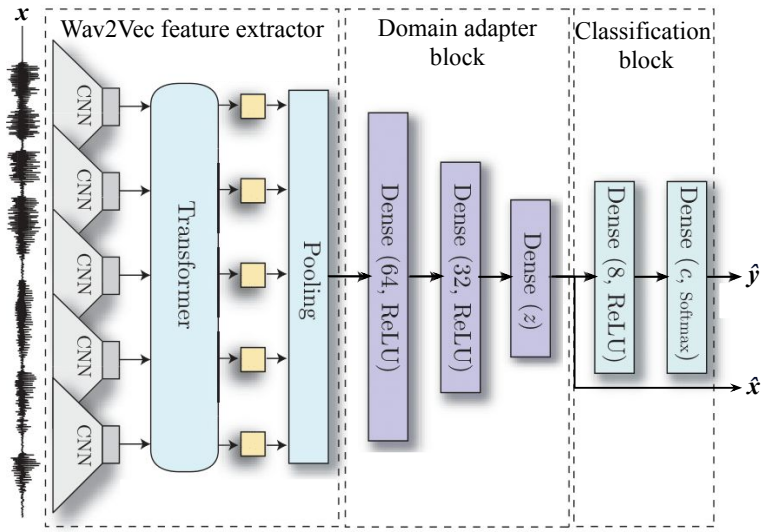


Figure 36: Schematic representation of the model.

of the model—where the entire TD training partition is used—and during the subsequent incremental adaptation steps, which rely exclusively on SS.

To construct **SS** set, we adopt two different strategies depending on the data domain. For the SD, samples are selected randomly, and the subset is expanded incrementally by retaining previously selected instances in each step. In contrast, for the TD, whose samples must be labeled by the user, we propose a selection strategy based on a modified Fixed-Center K-Means (FC-KMeans) clustering algorithm [151]. This method, described in detail in Algorithm 1, operates on the latent representation space of the unlabeled TD data, aiming to select a diverse and representative subset while minimizing redundancy among the chosen samples. Crucially, the selection process preserves previously chosen samples to ensure continuity and enable informed decision-making across incremental adaptation steps.

At the beginning, the selection method initializes k cluster centers, which determine the number of samples to select. In the first iteration, all centers are free, so the algorithm behaves like standard K-Means. It clusters the latent representations into k groups, then selects the representative samples closest to each centroid, taking them in their original representation space.

Algorithm 1 Selection method for representative samples.

Require:

TD Original space: $\mathbf{X}_{TD} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$,

TD Latent space: $\hat{\mathbf{X}}_{TD} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_N\}$,

Number of clusters k,

Latent representation of previously selected samples: $\hat{\mathbf{X}}'_{TD} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_f\}$

where $f < k$ and $\hat{\mathbf{X}}'_{TD} \subseteq \hat{\mathbf{X}}_{TD}$

Initialize $k - f$ remaining cluster centers: $\{\boldsymbol{\mu}_{f+1}, \ldots, \boldsymbol{\mu}_k\}$

repeat

for each point $\hat{\mathbf{x}}_i \in \hat{\mathbf{X}}_{TD}$ **do**

Assign $\hat{\mathbf{x}}_i$ to the nearest center: $c_i = \arg\min_j ||\hat{\mathbf{x}}_i - \boldsymbol{\mu}_j||^2$ for $j = 1, \ldots, k$

end for

for each non-fixed cluster $j = f + 1$ to k **do**

Update cluster center: $\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\hat{\mathbf{x}}_i \in C_j}^{N} \hat{\mathbf{x}}_i$

end for

until convergence

Updated selected samples: $\mathbf{X}'_{TD} = \left\{\mathbf{x}_i \mid i = \arg\min_{i' : \hat{\mathbf{x}}_{i'} \in C_j} \|\hat{\mathbf{x}}_{i'} - \boldsymbol{\mu}_j\|\right\}_{j=1}^{k}$

return Updated selected samples $\mathbf{X}'_{TD}$

In subsequent iterations, the representative samples previously selected ($\mathbf{X}'_{TD}$), along with their latent representations ($\hat{\mathbf{X}}'_{TD}$), are fixed as cluster centers. The selection method then updates the remaining centers to explore regions of the latent representation space that have not yet been covered. This allows the algorithm to progressively identify new representative samples that complement those already chosen. At each step, a new batch of raw audio inputs is selected based on their updated latent structure and added to the retraining set. This iterative process is repeated a limited number of times, aiming to adapt the model effectively while requiring only a small amount of user-labeled data.

The overall procedure for a single adaptation step within the proposed incremental semi-supervised speaker adaptation strategy is illustrated in Figure 37. This diagram summarizes the key components involved in the process: the generation of latent representations $\hat{X}_{TD}$ from the current unlabeled target

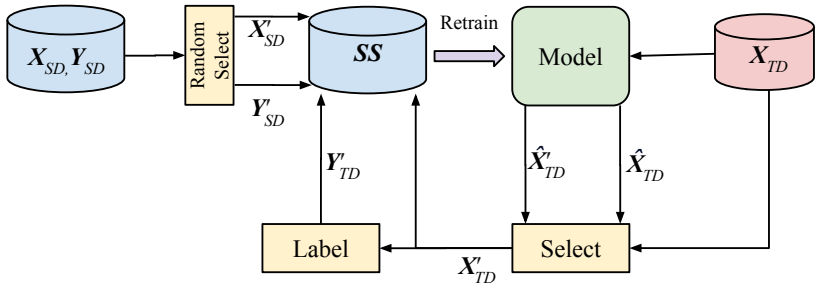


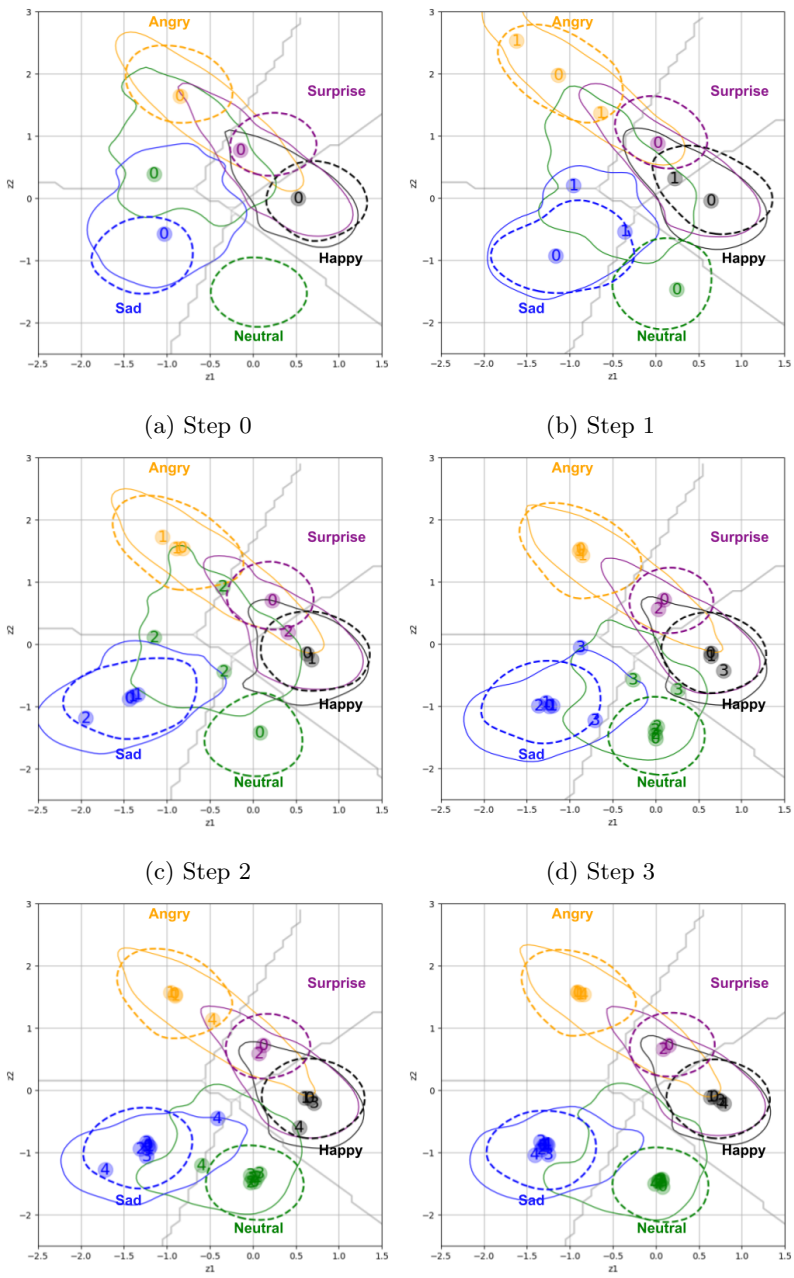
Figure 37: Overview of one step of the proposed incremental semi-supervised speaker adaptation strategy.

domain (TD) data, along with the use of previously selected samples $\hat{X}'_{TD}$; the selection of new informative samples and their annotations (X'_{TD}, Y'_{TD}); the combination with selected SD data (X'_{SD}, Y'_{SD}); and the subsequent retraining of the model. Each step incrementally improves the model's ability to recognize emotions from the new speaker while minimizing the annotation burden.

Figure 38 illustrates the evolution of the latent representation space during the incremental adaptation process. In this visualization, the last domain adaptation layer is configured to project the data into $\mathbb{R}^2$, meaning it consists of only two neurons. Samples from SD and TD are depicted using dashed and solid lines, respectively. Each real emotion class is identified by a distinct color, while the selected representative samples from TD are annotated with numbers indicating the iteration in which they were chosen. The classifier's decision boundaries are shown in gray, providing insight into how the model adjusts its internal representation to better separate emotion classes as the adaptation progresses.

Initially, samples from SD form well-defined and compact clusters in the latent representation space, reflecting the model's strong performance on seen speakers. In contrast, samples from TD appear dispersed and poorly aligned with the decision regions, which illustrates the performance degradation when applying the speaker-independent model to unseen speakers.

As the adaptation process begins, the selection method identifies representative samples from the TD distribution. While class balance is not explicitly enforced, the spatial structure of the latent representation space



(a) Step 0

(b) Step 1

(c) Step 2

(d) Step 3

(e) Step 4

(f) Step 5

Figure 38: Evolution of the latent representation space during incremental speaker adaptation.

naturally promotes a diverse selection. After each retraining step, the selected TD samples are progressively drawn closer to their respective class centroids due to the influence of the center loss term. This effect becomes particularly noticeable by iteration 5, where TD samples begin to form more coherent groupings, aligning better with the SD clusters and leading to more stable decision boundaries across domains.

Despite these improvements, some TD samples may remain misaligned or close to the decision boundaries, indicating that further iterations could still improve the adaptation. Nonetheless, this visualization highlights the effectiveness of the proposed strategy in gradually restructuring the latent representation space to accommodate a new speaker, while preserving the structure originally learned from SD.

5.2 Methodology

This section describes the experimental methodology employed to evaluate the proposed speaker adaptation strategy in SER. We detail the dataset partitioning, the incremental adaptation protocol, the selection methods, and the evaluation framework. The overall objective is to assess how effectively the model can personalize emotion recognition to new speakers with minimal labeled data, while retaining performance on previously seen speakers.

5.2.1 Experimental Setup

All experiments are conducted using the `SER` database, as introduced in Section 3.3.2. This dataset is partitioned into two groups: a source domain, composed of multiple speakers used to train a generic emotion recognition model, and a target domain, consisting of 20 held-out speakers used exclusively for speaker adaptation experiments. For both domains, the data is split into training and testing subsets, denoted as `SER_SD_TRAIN`, `SER_SD_TEST`, `SER_TD_TRAIN`, and `SER_TD_TEST`, respectively, following a 70%-30% ratio for training and testing. In the target domain, adaptation is performed one speaker at a time rather than

using all 20 speakers simultaneously.

Since Wav2Vec remains frozen throughout all experiments, the embeddings are precomputed for every utterance and used as the fixed input representations. This preprocessing step significantly reduces computational overhead during training and evaluation.

The initial model is trained on labeled utterances from the `SER_SD_TRAIN` subset, which includes multiple speakers. This model serves as the baseline for all subsequent adaptation experiments. It is trained to classify emotions into five categories: Sad, Angry, Neutral, Happy, and Surprise. The training process spans 200 epochs using a batch size of 64 and Adam optimizer. The loss function is a weighted sum of two components: $\mathcal{L}_{\text{CCE}}$, which guides the classification task, and center $\mathcal{L}_{\text{C}}$, which encourages intra-class compactness in the latent representation space. Specifically, the total loss is computed as $\mathcal{L} = \mathcal{L}_{\text{CCE}} + \lambda \cdot \mathcal{L}_{\text{C}}$, where $\lambda = 0.1$ This combination promotes both classification accuracy and robust embedding structure.

After training, the model is evaluated on the `SER_SD_TEST` subset to confirm its performance under speaker-dependent conditions. This pre-trained model also serves as the starting point for the incremental adaptation process applied to TD.

For the adaptation experiments, one speaker from the `SER_TD_TRAIN` subset is selected at a time. Although labels are available for this subset, we simulate a realistic scenario in which they are initially hidden and only revealed for a small number of representative samples suggested by the selection strategy. This approach allows us to emulate minimal supervision while progressively adapting the model to a new speaker.

At each iteration, a batch of 5 representative samples from the unlabeled target speaker is selected and labeled. This number is intentionally chosen to match the total number of emotion classes, encouraging the selection process to identify one representative sample per class. These samples are chosen using one of three selection methods: a purely random strategy, a prototype-based method known as Protodash, and our proposed FC-KMeans-based approach. Random selection simply picks new samples without any guidance. Protodash selects prototypes based on maximum relevance to the global distribution via

kernel-based optimization. However, since Protodash discards previously selected samples at each step and reselects over the cumulative set, it benefits from having seen all prior data—making it an upper-bound baseline rather than a fair incremental method.

After each adaptation step, the model is fine-tuned for 30 additional epochs using the newly labeled TD samples along with an equal number of randomly sampled labeled utterances from `SER_SD_TRAIN`. This balanced retraining strategy helps preserve knowledge from the source domain while gradually integrating new speaker-specific information.

All adaptation experiments are conducted in a four-dimensional latent representation space, produced by the domain adaptation block of the model. This space is crucial for the selection methods—Protodash and FC-KMeans—which rely on the spatial arrangement of embeddings to identify informative samples. The Random method, by contrast, does not depend on this latent structure.

The incremental adaptation continues for five steps, with 5 new representative samples labeled at each iteration, totaling 25 samples. This limit is chosen to reflect a practical annotation budget, as requesting more labeled data becomes increasingly burdensome for the user. Both our proposed FC-KMeans-based method and the Random strategy adhere to this constraint by incrementally adding 5 new samples per step, reusing all previously selected ones for training. In contrast, Protodash selects a new set of samples at each iteration based on the step's quota—5 in the first iteration, 10 in the second, 15 in the third, and so on. While it does not accumulate selections explicitly, it re-selects from the entire target dataset in each iteration using the updated latent space, potentially choosing different samples from previous steps. As a result, Protodash is effectively exposed to a growing number of labeled instances—up to 75 by the fifth step—despite only using the current selection for model adaptation.

5.2.2 Evaluation

The evaluation framework is based on classification accuracy (ACC), as previously mentioned, computed over two dataset partitions to assess both generalization and

knowledge retention. The SER_SD_TEST subset is used to evaluate the model's ability to maintain performance on previously seen speakers from the source domain, while the SER_TD_TEST subset measures generalization to the target speaker. These two metrics are monitored at each step of the incremental adaptation process to ensure that the model improves its performance on the target domain without significantly degrading accuracy on the source domain.

To account for variability due to random initialization and stochastic sample selection, all experiments are repeated 100 times per speaker. This results in a robust and statistically meaningful performance evaluation. To further validate the consistency of performance differences between selection methods, we apply the Wilcoxon signed-rank test. This non-parametric test does not assume normality and is well-suited for comparing paired results, such as classification accuracies obtained under different selection strategies but on the same experimental conditions. By analyzing the ranks of the differences, the Wilcoxon test offers a reliable way to determine whether improvements are statistically significant rather than due to chance.

This comprehensive evaluation protocol ensures the reliability of our results and underscores the effectiveness of the proposed selection method for incremental speaker adaptation.

5.3 Results and Discussion

This section presents and analyzes the results of the experiments designed to evaluate the effectiveness of the proposed incremental selection and adaptation (ISA) method. The main goal is to compare different sample selection strategies—Random, Protodash, and our proposed method—in terms of their ability to improve performance on the target domain (TD) while preserving accuracy on the source domain (SD).

To assess the robustness of each strategy, experiments are conducted under two distinct sampling conditions. The balanced scenario assumes that all emotion classes are equally represented in the unlabeled data pool, which favors class diversity. In contrast, the unbalanced scenario reflects a more realistic setting,

where the neutral class is overrepresented—posing a greater challenge for sample selection.

The analysis is organized in two parts. First, we present a detailed examination of the adaptation process for a two representative target speakers. These speakers were selected based on qualitative patterns observed across the full set, which are reported in Appendix C. This focused discussion allows us to highlight key behaviors and challenges encountered during the adaptation process.

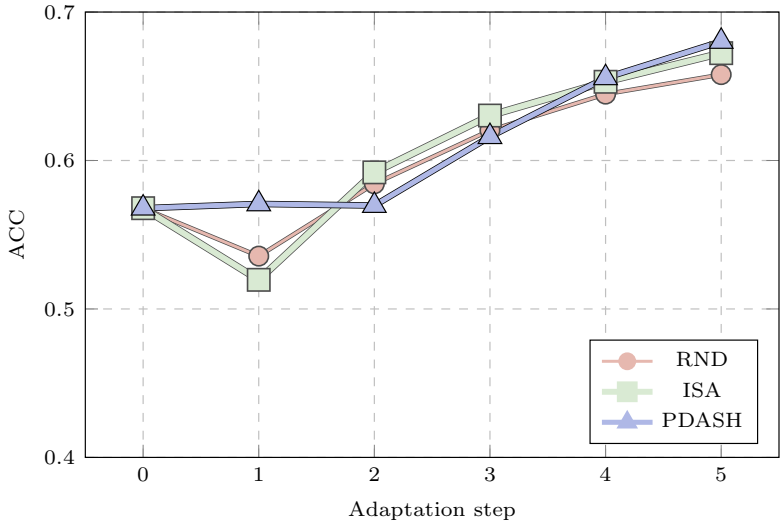
Second, we provide a global evaluation of each selection strategy across all target speakers. This includes comparisons of average performance and domain retention, as well as statistical significance tests to assess the consistency and reliability of the observed differences across conditions and budgets.

5.3.1 Speaker-wise Analysis

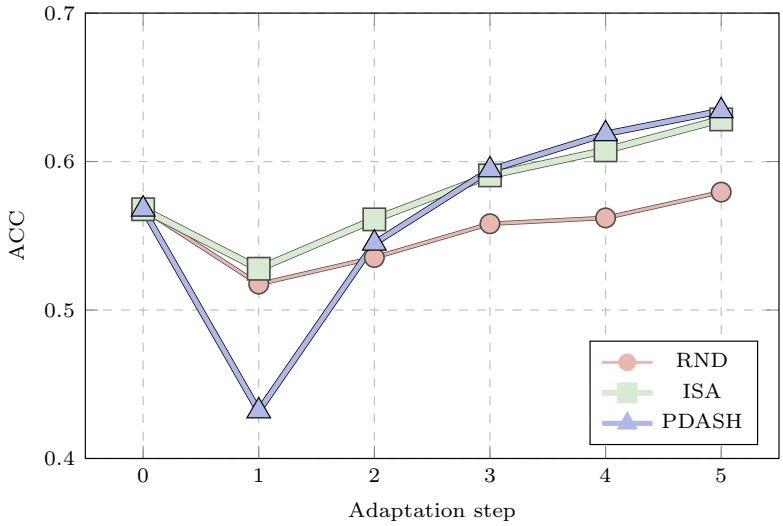
Figure 39 and Figure 40 illustrate the progression of average accuracy over 100 repetitions across five incremental adaptation steps for two representative target speakers: a Chinese speaker (Speaker 5) and an English speaker (Speaker 11). Both balanced and unbalanced scenarios are shown. Step 0 corresponds to the initial performance without any adaptation—serving as the common starting point for all methods under both balance conditions. As previously mentioned, results are reported using each speaker's test partition (`SER_TD_TEST`).

All methods show consistent improvements as more labeled data is incorporated, confirming both the benefits of incremental adaptation and the generalization challenges models face when adapting to new speakers. The Protodash selection strategy typically achieves the highest accuracy (ACC) values at the final step, closely followed by our proposed method, ISA—a difference that becomes less pronounced under unbalanced conditions. In contrast, the Random selection also shows improvements over the initial step (step 0), but consistently lags behind the other two strategies at the final stage.

Although Protodash yields the highest ACC values, it is important to note that by step 5, this method has used 75 labeled samples, whereas ISA has required only 25. This highlights the efficiency of ISA in achieving competitive results with

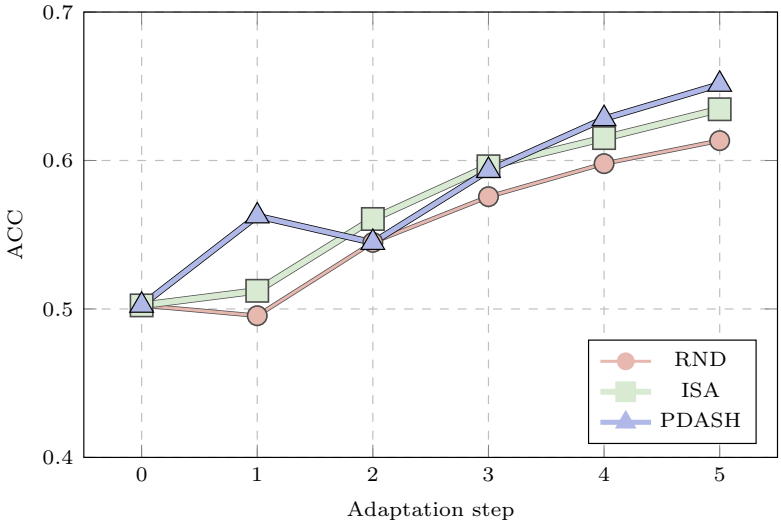


(a) Balanced conditions.

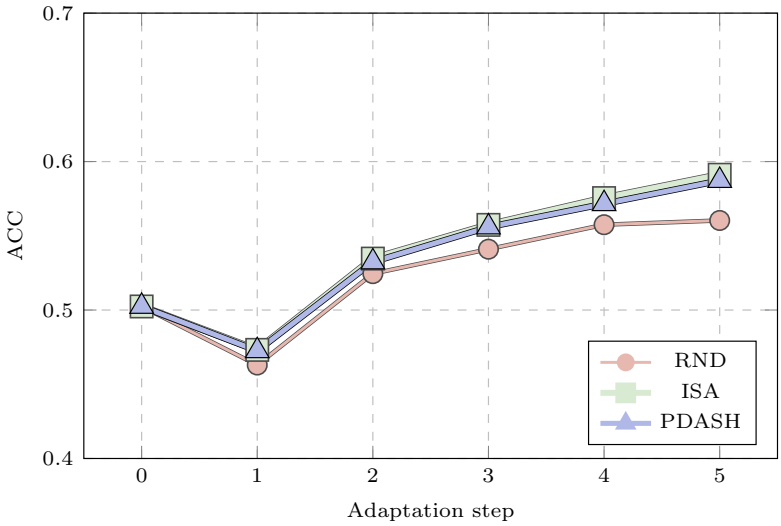


(b) Unbalanced conditions.

Figure 39: Incremental adaptation performance for Speaker 5 (Chinese).



(a) Balanced conditions.



(b) Unbalanced conditions.

Figure 40: Incremental adaptation performance for Speaker 11 (English).

significantly fewer labeled samples.

A fairer comparison between ISA and Protodash can be made by contrasting ISA at step 5 with Protodash at step 3, where both methods have used a comparable number of labeled samples (25 for ISA and 30 for Protodash). In this comparison, ISA shows a clear advantage in the ACC values achieved for both speakers and under both balance conditions, reinforcing the effectiveness of the proposed approach.

A key aspect to consider is the difference in maximum ACC values reached under balanced and unbalanced scenarios. In the latter, all methods are affected by the unequal class distribution, especially when they fail to select representative samples from all classes. In the case of Random selection, this issue is even more problematic, as it fully depends on the distribution of the available unlabeled data. In contrast, ISA mitigates this effect by selecting regions in the latent space that are more likely to cover class diversity, even under imbalance.

Finally, a drop in accuracy is observed at step 1 compared to the baseline (step 0). This may be due to the fact that retraining with only 10 samples is too aggressive and temporarily destabilizes the model. However, this trend reverses in the subsequent steps, where the model progressively improves as more labeled samples are introduced.

5.3.2 Global Performance Trends

To further support and generalize the previous findings, Figure 41 presents the average accuracy progression across all 20 target speakers under both balanced and unbalanced conditions. These aggregated curves reflect global trends across the full evaluation set and provide a more comprehensive view of each strategy's behavior.

At a global level, the patterns observed are highly consistent with the individual speaker analysis presented earlier. All strategies improve over time, and Random selection remains the least effective. Protodash tends to achieve the highest final ACC, but again at the cost of a significantly larger labeled sample budget. ISA, in contrast, shows a comparable performance trajectory, particularly

under the constrained budget, confirming its efficiency and competitiveness. These overall trends help to generalize many of the conclusions drawn from the case studies and validate them across a broader population of target speakers.

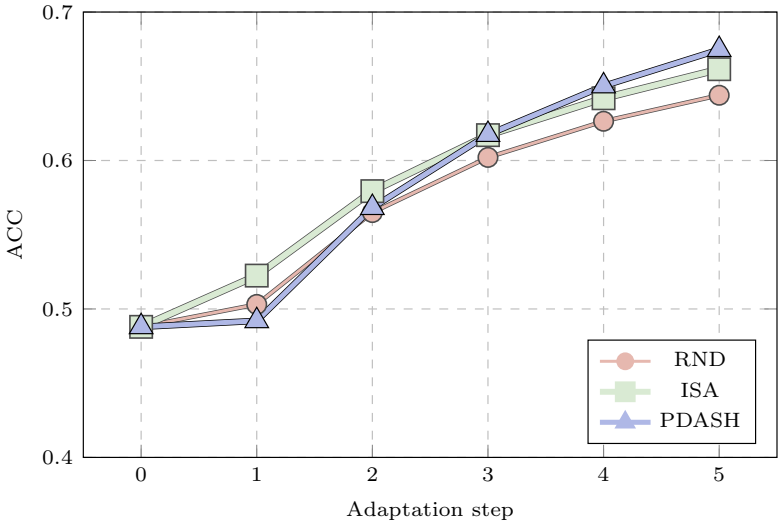
Figure 42 complements this view by showing the evolution of average accuracy in both TD and SD using ISA. As expected, the TD benefits from incremental adaptation, whereas the SD exhibits a drop in accuracy. This degradation is most pronounced at step 1, suggesting that the initial retraining with only 10 samples may destabilize the model, similar to the effect observed earlier in the TD. However, as more labeled samples are incorporated, SD performance begins to recover, indicating that the adaptation process gradually stabilizes.

Additionally, we observe a notable difference in the final SD accuracy (step 5) between balanced and unbalanced conditions. In the unbalanced setting, the maximum ACC reached in SD is lower than in the balanced one. This suggests that the quality and diversity of the selected TD samples have a direct impact not only on adaptation performance but also on the model's ability to preserve prior knowledge. In other words, effective sample selection in TD plays a dual role: it facilitates adaptation and mitigates forgetting.

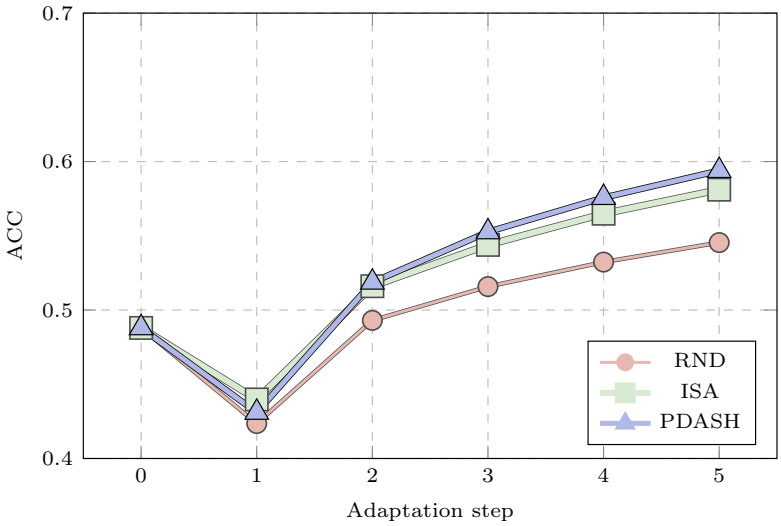
To complement the visual analysis, Table 10 summarizes the average accuracy (ACC) across the 20 target speakers for both the target domain and the source domain, under different sample selection strategies and conditions. The table includes pairwise statistical comparisons using the non-parametric Wilcoxon signed-rank test. The symbols $<$, $=$, and $>$ indicate whether the difference is statistically significant in favor of, equal to, or against the reference method ($p < 0.05$).

Results are reported for two key adaptation steps: step 3 and step 5, which correspond to 15 and 25 labeled samples for ISA and Random, and to 30 and 75 labeled samples for Protodash.

The results reinforce several of the previously discussed insights. In the TD, ISA consistently outperforms random selection in both scenarios, with statistically significant differences evident even under the more constrained labeling budget (15 samples). Protodash achieves the highest ACC when given access to 75 labeled samples, although this advantage is not always statistically significant

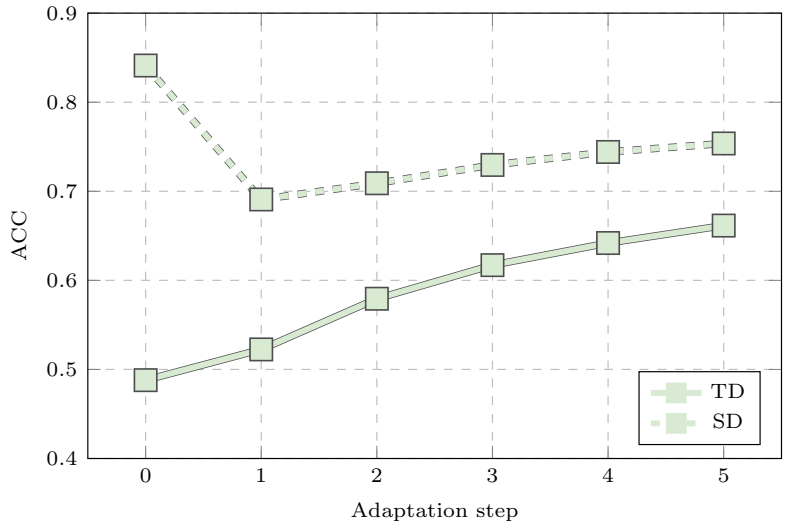


(a) Balanced conditions.

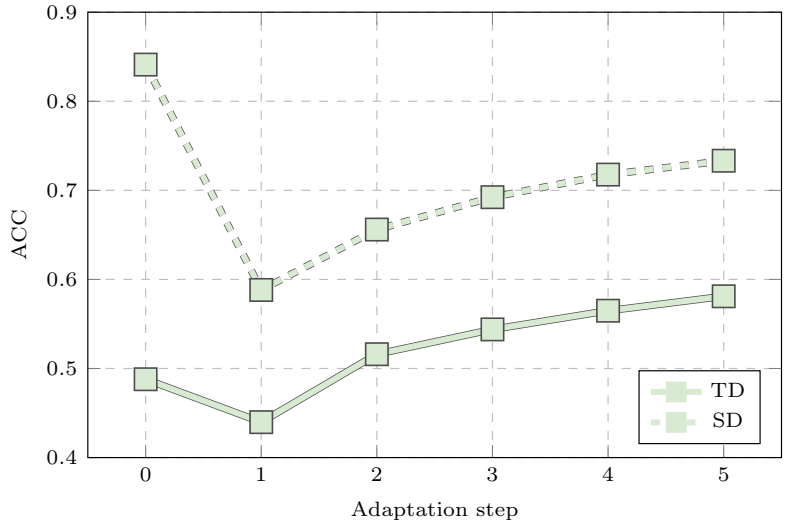


(b) Unbalanced conditions.

Figure 41: Mean accuracy across all 20 target domain speakers.



(a) Balanced conditions.



(b) Unbalanced conditions.

Figure 42: Mean accuracy for all 20 speakers progression in both SD and TD using ISA.

Table 10: Average accuracy scores across 20 target speakers under different sample selection strategies and balance conditions.

		init	RND		ISA		PDASH	RND		ISA		PDASH
Step		0			3					5		
Samples		0	15		15		30	25		25		75
TD	(U)	0.49	0.52	<	0.54	=	0.55	0.55	<	0.58	<	0.59
	(B)		0.60	<	0.62	=	0.62	0.64	<	0.66	<	0.68
SD	(U)	0.84	0.69	=	0.69	=	0.69	0.73	=	0.73	=	0.73
	(B)		0.72	<	0.73	>	0.72	0.74	<	0.75	>	0.74

when compared to ISA at step, supporting the argument that its performance gain is partially attributable to its greater labeling budget.

Across both sample budgets, balanced conditions yield higher ACC than unbalanced ones for all strategies, reaffirming the importance of class diversity in adaptation. Despite this, ISA maintains competitive results even under the skewed distribution, highlighting its robustness.

In the SD, as expected, a gradual drop in ACC is observed with adaptation. This degradation is more pronounced under unbalanced conditions, suggesting that the nature of selected TD samples can inadvertently affect the preservation of prior knowledge. Notably, under balanced conditions, ISA demonstrates better retention of SD accuracy compared to Protodash in several configurations, with statistically significant differences in some cases.

Overall, these results validate ISA as an efficient and effective alternative to more data-intensive strategies like Protodash. Its ability to maintain solid adaptation performance while preserving prior knowledge—especially under realistic, unbalanced conditions—makes it a strong candidate for scalable and cost-effective domain adaptation in emotion recognition.

5.4 Strengths and Limitations of the Incremental Strategy

The incremental adaptation strategy evaluated throughout this chapter offers several notable strengths that make it well-suited for emotion recognition systems

facing speaker variability challenges. A key advantage is its ability to progressively tailor the model to new target speakers using a very limited number of labeled samples. As discussed in Appendix D, subsets of as few as 15 to 25 carefully selected samples can achieve target domain accuracy comparable to training with the full dataset. Our proposed ISA method represents a step in this direction by intelligently identifying these representative samples for effective adaptation.

Another important strength lies in the modularity and transparency of the incremental process. By updating the model in discrete steps, it allows controlled incorporation of new information and offers insight into how the model evolves over time. Additionally, selection methods such as ISA effectively balance adaptation to the new speaker with preservation of performance on the original source domain, thus mitigating catastrophic forgetting — a critical requirement for deploying robust models in multi-speaker or evolving environments.

Despite these advantages, the approach also has limitations. Its success depends heavily on the quality of the sample selection strategy. As show, naive random selection consistently underperforms compared to informed methods like ISA or Protodash, indicating that effective sample representativeness in the latent space is crucial.

Another limitation is the instability observed in the initial adaptation step, especially under unbalanced data conditions. When retraining occurs with very few and potentially non-representative samples, performance can temporarily decline before improving in later steps. This highlights the importance of considering class distribution and sample diversity when designing adaptation strategies.

In summary, the incremental strategy provides a scalable and sample-efficient pathway for speaker adaptation in speech emotion recognition. When combined with well-designed selection methods and applied under reasonable assumptions about class distribution, it offers a valuable alternative to full retraining while maintaining competitive performance.

Chapter 6:
Conclusions and Future Work

6.1 General Conclusions

In this section, the main conclusions of the thesis are presented following the structure of its three core contributions. Each contribution addresses a distinct challenge in the field of acoustic monitoring and sound event classification and detection within driving environments. To highlight their individual advances, the conclusions are discussed separately. A final summary then integrates the overarching insights and reflects on the overall impact of the work.

The first contribution introduced a domain-specific framework for acoustic monitoring in driving scenarios, establishing a comprehensive taxonomy of sound events with a particular focus on auditory distractors. While previous studies had examined these factors in isolation, this work provided the first unified taxonomy tailored to the driving context. Complementing this framework, a curated dataset was developed by combining synthetic and real-world audio samples, offering a valuable resource for both academic research and applied safety systems. The usefulness of this dataset was validated through the training and evaluation of detection models that achieved good performance in real-world driving conditions, supporting its relevance and practical applicability.

The second contribution focused on the design and evaluation of an efficient sound event detection model based on the YOLO architecture, adapted for processing audio spectrograms. Experimental results showed that this YOLO-inspired model surpassed a CRNN baseline in detection accuracy, confirming the viability of applying image-based convolutional techniques to acoustic data. This approach leveraged structural similarities between visual and acoustic representations, highlighting the potential of cross-domain architectural adaptations.

In addition to accuracy, the study emphasized computational efficiency—an

essential requirement for deployment in automotive edge devices. The model demonstrated a favorable trade-off between performance and resource usage, enabling real-time operation with low latency and reduced power consumption. These characteristics make it a strong candidate for integration into practical, on-board acoustic monitoring systems.

The third contribution addressed the issue of model generalization across varying acoustic conditions through an incremental, semi-supervised domain adaptation strategy. This method enables progressive model refinement with minimal labeled data by selectively incorporating representative samples from new environments or speakers. Applied to emotion recognition tasks, the approach demonstrated significant improvements in adaptation performance under both balanced and unbalanced data conditions. These findings reinforce the importance of adaptive learning strategies to ensure robustness in dynamic, real-world scenarios, such as those encountered in driving contexts.

In summary, this thesis presents a coherent and innovative approach to improving acoustic monitoring systems for driving safety. Through the integration of a specialized taxonomy and dataset, a lightweight and accurate detection model, and a robust adaptation strategy, the work advances the state of the art in multiple dimensions. Collectively, these contributions offer scalable, efficient, and adaptive solutions that pave the way toward safer and more intelligent transportation systems.

6.2 Scientific Research Result

The work carried out in this thesis has led to the publication of several peer-reviewed journal articles, each contributing in different aspects of the research and collectively supporting its core scientific contributions. These contributions, as outlined in the Chapter 1, include: (i) a framework for acoustic monitoring in driving environments, (ii) a CNN-based approach for efficient sound event detection, and (iii) an incremental domain adaptation strategy for sound events.

The article “A safety-oriented framework for sound event detection in driving

scenarios", published in Applied Acoustics (2023) [9], laid the foundation for the first major contribution of this thesis: A Framework for Acoustic Monitoring in Driving Environments. This work introduced a domain-specific taxonomy focused on sound events relevant to driving scenarios, emphasizing auditory distractions and safety-critical cues. The publication also described the creation of a dataset combining synthetic and real audio recordings from driving environments, which became a crucial resource for subsequent model training and evaluation. Additionally, the paper proposed a novel sound event detection method based on a YOLO-inspired architecture that relies exclusively on convolutional layers. This approach served as an alternative to recurrent models for sound event detection and demonstrated the viability of convolutional architectures in this domain.

Building upon that foundation, the article "Edge computing for driving safety: evaluating deep learning models for cost-effective sound event detection", published in The Journal of Supercomputing (2025) [16], aligns closely with the second contribution of this thesis: CNN-Based Approach for Efficient Sound Event Detection. This study focused on evaluating lightweight deep learning architectures optimized for real-time inference on edge devices. Specifically, it benchmarked a YOLO-inspired model and a CRNN baseline across multiple hardware platforms and different levels of model compression, demonstrating the feasibility of deploying efficient sound event detection systems in automotive environments without sacrificing accuracy or latency.

The third article, "Exploring selective sampling bounds for speaker-dependent speech emotion recognition", published in Ideas en Ciencias de la Ingeniería (2025) [17], is primarily associated with the third contribution: Incremental Domain Adaptation Strategy for Sound Events. Although this work focused on speech emotion recognition, it addressed the broader challenge of adapting models to new domains—in this case, new speakers—with limited labeled data. The results demonstrated that appropriate sample selection and transfer learning techniques can substantially improve performance in novel acoustic contexts, reinforcing the thesis's emphasis on robust and adaptive modeling strategies.

Additionally, a recent article currently under review, titled "An Incremental Selection Method for Semi-supervised Speaker Adaptation in Speech Emotion

Recognition," is directly based on the third contribution: Incremental Domain Adaptation Strategy for Sound Events. This work presents the incremental adaptation framework described in Chapter 5 of this thesis, leveraging a modified k-means algorithm to iteratively select representative samples for aligning a source domain to new speakers. By progressively refining the model with selected samples, it achieves improved generalization under both balanced and unbalanced data conditions. The reported results demonstrate superior or comparable performance to state-of-the-art non-incremental methods, reinforcing the practical value of the proposed incremental strategy for robust model adaptation in challenging acoustic environments.

Finally, various conference papers were published during the development of this research. While not described in detail here, these publications helped disseminate early results and provided experimental feedback that influenced the final contributions of the thesis.

6.3 Future Work

Building upon the foundations established in this thesis, several promising directions emerge for future research.

First, although the developed dataset proved sufficient for training models capable of interpreting the acoustic landscape of driving environments, its expansion remains a key area for improvement. In particular, increasing the volume and diversity of real-world audio recordings could enhance the robustness of evaluations, which—as discussed in previous chapters—are heavily influenced by the structure and representativeness of the datasets. A richer corpus would support more comprehensive benchmarking and facilitate the development of models better suited to real-world deployment. Additionally, increasing the realism and variability of synthetic data through improved simulation techniques and acoustic modeling can provide more challenging and representative training scenarios, helping to bridge the gap between synthetic and real acoustic environments.

Furthermore, the proposed taxonomy of acoustic events could evolve towards

more granular and contextually sensitive forms. The current structure is based on empirical observations and specialized literature, but expanding towards taxonomies specific to different vehicle types—such as electric, commercial, or public transport vehicles—would enable the development of models better adapted to each use scenario. Even more promising is the development of dynamic and multimodal taxonomies, which integrate data from visual, acoustic, and vehicle sensors, allowing models to adjust their categories depending on the type of route, driver behavior, or environmental conditions.

Second, while the YOLO-based model proposed in this thesis has shown to be an effective approach for sound event detection, further adaptations could help tailor the architecture more specifically to the characteristics of audio data. For instance, future iterations could eliminate the frequency-axis component of the output representation, as sound events in spectrograms are typically localized temporally rather than spatially across frequencies. Additionally, integrating audio-specific data augmentation techniques into the training process would likely improve generalization. Another key avenue involves enabling the model to take advantage of pre-trained audio representations and to support diverse forms of supervision, including weak labels and unlabeled data. These enhancements would align the architecture with the latest trends in audio-based deep learning.

Another important line of future work involves developing a working prototype that incorporates the proposed detection model into a real-time system, with special attention to user experience. Such a system could serve as a practical tool for in-vehicle acoustic monitoring, contributing directly to road safety and enabling validation in real driving scenarios.

Regarding the third contribution, future efforts should explore extending the proposed incremental domain adaptation strategy beyond the speech emotion recognition setting. Applying this approach to broader sound event detection tasks—such as adapting to new types of auditory distractors or environmental noise—could yield significant benefits in terms of model robustness and adaptability. Moreover, investigating the joint modeling of emotion recognition and event detection may uncover synergies between these tasks. A multi-task or unified framework could potentially leverage overlapping acoustic cues, enhancing

performance in both domains and providing a more holistic understanding of the auditory context in driving environments.

In summary, the thesis opens several promising research directions, both for refining the technical approaches developed and for broadening their application in real-world intelligent transport systems.

References

[1] M. Neri, F. Battisti, A. Neri, M. Carli, Sound Event Detection for Human Safety and Security in Noisy Environments, IEEE Access 10 (2022). doi:10.1109/ACCESS.2022.3231681.

[2] R. M. Alsina-Pagès, J. Navarro, F. Alías, M. Hervás, homeSound: Real-Time Audio Event Detection Based on High Performance Computing for Behaviour and Surveillance Remote Monitoring, Sensors 17 (4) (2017). doi:10.3390/s17040854.

[3] A. Gupta, A. Anpalagan, L. Guan, A. S. Khwaja, Deep Learning for Object Detection and Scene Perception in Self-Driving Cars: Survey, Challenges, and Open Issues, Array 10 (2021) 100057. doi:10.1016/j.array.2021.100057.

[4] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, D. Levi, Driver Gaze Tracking and Eyes Off the Road Detection System, IEEE Transactions on Intelligent Transportation Systems 16 (4) (2015) 2014–2027. doi:10.1109/TITS.2015.2396031.

[5] S. Jiang, Z. Guo, S. Zhao, H. Wang, W. Jing, CE-GAN: A Camera Image Enhancement Generative Adversarial Network for Autonomous Driving, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), 2022, pp. 1–6. doi:10.1109/DSAA54385.2022.10032427.

[6] P. Marcillo, A. L. Valdivieso Caraguay, M. Hernandez-Alvarez, A Systematic Literature Review of Learning-Based Traffic Accident Prediction Models Based on Heterogeneous Sources, Applied Sciences 12 (9) (2022). doi:10.3390/app12094529.

[7] H. H. Hussein, O. Karan, S. Kurnaz, Enhancing Driving Control via Speech Recognition Utilizing Influential Parameters in Deep Learning Techniques, Electronics 14 (3) (2025). doi:10.3390/electronics14030496.

[8] D. Ivanko, D. Ryumin, A. Axyonov, A. Kashevnik, Speaker-Dependent Visual Command Recognition in Vehicle Cabin: Methodology and Evaluation, in: A. Karpov, R. Potapova (Eds.), Speech and Computer, 2021, pp. 291–302. doi:10.1007/978-3-030-87802-3_27.

[9] C. Castorena, M. Cobos, J. Lopez-Ballester, F. J. Ferri, A Safety-Oriented Framework for Sound Event Detection in Driving Scenarios, Applied Acoustics 215 (2023). doi:10.1016/j.apacoust.2023.109719.

[10] Z. Halim, M. Rehan, On Identification of Driving-Induced Stress Using Electroencephalogram Signals: A Framework Based on Wearable Safety-Critical Scheme and Machine Learning, Information Fusion 53 (2020) 66–79. doi:10.1016/j.inffus.2019.06.006.

[11] S. Zepf, J. Hernandez, A. Schmitt, W. Minker, R. W. Picard, Driver Emotion Recognition for Intelligent Vehicles: A Survey, ACM Comput. Surv. 53 (3) (jul 2020). doi:10.1145/3388790.

[12] A. Maccagno, A. Mastropietro, U. Mazziotta, M. Scarpiniti, Y.-C. Lee, A. Uncini, A CNN Approach for Audio Classification in Construction Sites, in: Progresses in Artificial Intelligence and Neural Systems, Springer Singapore, Singapore, 2021, pp. 371–381. doi:10.1007/978-981-15-5093-5_33.

[13] H. Zhu, J. Yan, A Deep Learning Based Sound Event Location and Detection Algorithm Using Convolutional Recurrent Neural Network, in: 2022 International Conference on Computer, Information and Telecommunication Systems (CITS), 2022, pp. 1–6. doi:10.1109/CITS55221.2022.9832991.

[14] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, Advances in Neural Information Processing Systems 33 (2020) 12449–12460. doi:10.48550/arXiv.2006.11477.

[15] J. Wagner, A. Triantafyllopoulos, H. Wierstorf, M. Schmitt, F. Burkhardt, F. Eyben, B. W. Schuller, Dawn of the Transformer Era in Speech Emotion Recognition: Closing the Valence Gap, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 10745–10759. doi:10.1109/TPAMI.2023.3263585.

[16] C. Castorena, J. Lopez-Ballester, J. A. D. Rus, M. Cobos, F. J. Ferri, Edge Computing for Driving Safety: Evaluating Deep Learning Models for Cost-Effective Sound Event Detection, J Supercomput 288 (2025). doi:10.1007/s11227-024-06796-1.

[17] C. Castorena, J. Lopez-Ballester, F. Ferri, M. Cobos, Exploring Selective Sampling Bounds for Speaker-Dependent Speech Emotion Recognition, Ideas en Ciencias de la Ingeniería 3 (1) (2025) 48–58. doi:10.36677/rici.v3i1.25132.

[18] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. P. Bello, Robust Sound Event Detection in Bioacoustic Sensor Networks, PLoS ONE 14 (10) (2019). doi:10.1371/journal.pone.0214168.

[19] I. Mykhailichenko, H. Ivashchenko, O. Barkovska, O. Liashenko, Application of Deep Neural Network for Real-Time Voice Command Recognition, in: 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), 2022, pp. 1–4. doi:10.1109/KhPIWeek57572.2022.9916473.

[20] M. D. Zbancioc, R. Butnaru, S. M. Feraru, Recognition of Voice Commands using CNN for Romanian Language, in: 2022 E-Health and Bioengineering Conference (EHB), 2022, pp. 1–4. doi:10.1109/EHB55594.2022.9991322.

[21] N. Ndou, R. Ajoodha, A. Jadhav, Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches, in: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, p. 1. doi:10.1109/IEMTRONICS52119.2021.9422487.

[22] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, A. Q. Ohi, A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities, IEEE Access 9 (2021) 79236–79263. doi:10.1109/ACCESS.2021.3084299.

[23] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, T. Virtanen, Low-complexity acoustic scene classification in DCASE 2022 Challenge (2022). doi:10.48550/arXiv.2206.03835.

[24] A. F. Romero, A. D. Orjuela-Cañón, Respiratory Sounds Classification Employing a Multi-Label Approach, in: 2021 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI), 2021, pp. 1–5. doi:10.1109/ColCACI52978.2021.9469042.

[25] T. A. Onisha, J. Kim, J. Seol, Multi-Label Sound Classification Using Deep Learning Models, in: 2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA), 2024, pp. 129–134. doi:10.1109/SERA61261.2024.10685563.

[26] H. Phan, T. N. T. Nguyen, P. Koch, A. Mertins, Polyphonic Audio Event Detection: Multi-Label or Multi-Class Multi-Task Classification Problem?, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 8877–8881. doi:10.1109/ICASSP43922.2022.9746402.

[27] T. K. Chan, C. S. Chin, A Comprehensive Review of Polyphonic Sound Event Detection, IEEE Access 8 (2020). doi:10.1109/ACCESS.2020.2999388.

[28] A. Mesaros, T. Heittola, T. Virtanen, M. D. Plumbley, Sound Event Detection: A Tutorial, IEEE Signal Processing Magazine 38 (2021). doi:10.1109/MSP.2021.3090678.

[29] N. Turpault, R. Serizel, A. P. Shah, J. Salamon, Sound Event Detection in Domestic Environments with Weakly Labeled Data and Soundscape

Synthesis, in: Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), 2019, p. 1.

[30] V. Arroyo, J. J. Valero-Mas, J. Calvo-Zaragoza, A. Pertusa, Neural Audio-To-Score Music Transcription For Unconstrained Polyphony Using Compact Output Representations, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 4603–4607. doi:10.1109/ICASSP43922.2022.9746239.

[31] T. Fernando, S. Sridha, S. Denman, H. Ghaemmaghami, C. Fookes, Robust and Interpretable Temporal Convolution Network for Event Detection in Lung Sound Recordings, IEEE Journal of Biomedical and Health Informatics 26 (2022). doi:10.1109/JBHI.2022.3144314.

[32] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, M. Plakal, The Benefit of Temporally-Strong Labels in Audio Event Classification, in: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 366–370. doi:10.1109/ICASSP39728.2021.9414579.

[33] I. Martín-Morató, A. Mesaros, Strong Labeling of Sound Events Using Crowdsourced Weak Labels and Annotator Competence Estimation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 31 (2023) 902–914. doi:10.1109/TASLP.2022.3233468.

[34] J. Vijay, Y. Kawanishi, Generating Pseudo-Strong Labels from Weak Labels for Distributed Multi-Microphone Sound Event Detection, in: Pattern Recognition, Springer Nature Switzerland, 2025, pp. 98–113. doi:10.1007/978-3-031-78192-6_7.

[35] K. Zaman, M. Sah, C. Direkoglu, M. Unoki, A Survey of Audio Classification Using Deep Learning, IEEE Access 11 (2023) 106620–106649. doi:10.1109/ACCESS.2023.3318015.

[36] A. Dang, T. H. Vu, J.-C. Wang, A Survey of Deep Learning for Polyphonic Sound Event Detection, in: 2017 International Conference on Orange Technologies (ICOT), 2017, pp. 75–78. doi:10.1109/ICOT.2017.8336092.

[37] D. Pramanick, H. Ansar, H. Kumar, P. S, R. Tengshe, B. Fatimah, Deep Learning Based Urban Sound Classification and Ambulance Siren Detector Using Spectrogram, in: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1–6. doi:10.1109/ICCCNT51525.2021.9579778.

[38] C. Castorena, F. J. Ferri, M. Cobos, On the Performance of Deep Learning Models for Respiratory Sound Classification Trained on Unbalanced Data, in: Pattern Recognition and Image Analysis, 2022, pp. 143–155. doi:10.1007/978-3-031-04881-4_12.

[39] R. M. Souza, E. G. Nascimento, U. A. Miranda, W. J. Silva, H. A. Lepikson, Deep Learning for Diagnosis and Classification of Faults in Industrial Rotating Machinery, Computers & Industrial Engineering 153 (2021) 107060. doi:10.1016/j.cie.2020.107060.

[40] D. Bonet-Solà, R. M. Alsina-Pagès, A Comparative Survey of Feature Extraction and Machine Learning Methods in Diverse Acoustic Environments, Sensors 21 (4) (2021). doi:10.3390/s21041274.

[41] Çağdaş Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, S. Krstulović, A Framework for the Robust Evaluation of Sound Event Detection, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 61–65. doi:10.1109/ICASSP40776.2020.9052995.

[42] Y. Zhang, B. Li, H. Fang, Q. Meng, Spectrogram Transformers for Audio Classification, in: 2022 IEEE International Conference on Imaging Systems and Techniques (IST), 2022, pp. 1–6. doi:10.1109/IST55454.2022.9827729.

[43] K. Li, Y. Song, L.-R. Dai, I. McLoughlin, X. Fang, L. Liu, AST-SED: An Effective Sound Event Detection Method Based on Audio Spectrogram Transformer, in: 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10096853.

[44] C. Ick, B. McFee, Sound Event Detection in Urban Audio with Single and Multi-Rate PCEN, in: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 880–884. doi:10.1109/ICASSP39728.2021.9414697.

[45] S. Venkatesh, D. Moffat, E. R. Miranda, You Only Hear Once: A YOLO-like Algorithm for Audio Segmentation and Sound Event Detection, Applied Sciences 12 (7) (2022). doi:10.3390/app12073293.

[46] L. Durak, O. Arikan, Short-time Fourier transform: Two Fundamental Properties and an Optimal Implementation, IEEE Transactions on Signal Processing 51 (5) (2003) 1231–1242. doi:10.1109/TSP.2003.810293.

[47] D. Ćirić, Z. Perić, J. Nikolić, N. Vučić, Audio Signal Mapping into Spectrogram-Based Images for Deep Learning Applications, in: 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021, pp. 1–6. doi:10.1109/INFOTEH51037.2021.9400698.

[48] S. Molau, M. Pitz, R. Schluter, H. Ney, Computing Mel-frequency Cepstral Coefficients on the Power Spectrum, in: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP), Vol. 1, 2001, pp. 73–76. doi:10.1109/ICASSP.2001.940770.

[49] S. Wyatt, D. Elliott, A. Aravamudan, C. E. Otero, L. D. Otero, G. C. Anagnostopoulos, A. O. Smith, A. M. Peter, W. Jones, S. Leung, E. Lam, Environmental Sound Classification with Tiny Transformers in Noisy Edge Environments, in: 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 309–314. doi:10.1109/WF-IoT51360.2021.9596007.

[50] M. Massoudi, S. Verma, R. Jain, Urban Sound Classification using CNN, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 583–589. doi:10.1109/ICICT50816.2021.9358621.

[51] A. A. Khamees, H. D. Hejazi, M. Alshurideh, S. A. Salloum, Classifying Audio Music Genres Using CNN and RNN, in: Advanced Machine Learning Technologies and Applications, Springer International Publishing, Cham, 2021, pp. 315–323. doi:10.1007/978-3-030-69717-4_31.

[52] N. Nakngoen, P. Pongboriboon, N. Inthanop, J. Akharachaisirilap, T. Woodward, N. Teerasuttakorn, Drone Classification Using Gated Recurrent Unit, in: IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, 2023, pp. 1–4. doi:10.1109/IECON51785.2023.10312193.

[53] W. Zhu, M. Omar, Multiscale Audio Spectrogram Transformer for Efficient Audio Classification, in: 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10096513.

[54] S. Sadhu, D. He, C.-W. Huang, S. H. Mallidi, M. Wu, A. Rastrow, A. Stolcke, J. Droppo, R. Maas, Wav2vec-C: A Self-supervised Model for Speech Representation Learning (2021). doi:10.48550/arXiv.2103.08393.

[55] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, K. Wilson, CNN architectures for large-scale audio classification, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 131–135. doi:10.1109/ICASSP.2017.7952132.

[56] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[57] N. Ketkar, J. Moolayil, Convolutional neural networks, in: Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch,

Apress, Berkeley, CA, 2021, pp. 197–242. doi:10.1007/978-1-4842-5364-9_6.

[58] Zhou, Chellappa, Computation of optical flow using a neural network, in: IEEE 1988 International Conference on Neural Networks, 1988, pp. 71–78 vol.2. doi:10.1109/ICNN.1988.23914.

[59] A. Ajit, K. Acharya, A. Samanta, A Review of Convolutional Neural Networks, in: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1–5. doi:10.1109/ic-ETITE47903.2020.049.

[60] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, Vol. 37 of Proceedings of Machine Learning Research, PMLR, Lille, France, 2015, pp. 448–456.

[61] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[62] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation (2014). doi:10.48550/arXiv.1406.1078.

[63] N. Asatani, T. Kamiya, S. Mabu, S. Kido, Classification of Respiratory Sounds Using Improved Convolutional Recurrent Neural Network, Computers & Electrical Engineering 94 (2021) 107367. doi:10.1016/j.compeleceng.2021.107367.

[64] H. Yadav, P. Shah, N. Gandhi, T. Vyas, A. Nair, S. Desai, L. Gohil, S. Tanwar, R. Sharma, M. Verdes, M. S. Raboaca, CNN and Bidirectional GRU-Based Heartbeat Sound Classification Architecture for Elderly People, Mathematics 11 (6) (2023). doi:10.3390/math11061365.

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, I. Polosukhin, Attention is All You Need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, p. 1.

[66] I. Martín-Morató, M. Harju, P. Ahokas, A. Mesaros, Training Sound Event Detection with Soft Labels from Crowdsourced Annotations, in: 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10095504.

[67] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A Discriminative Feature Learning Approach for Deep Face Recognition, in: Computer Vision – ECCV 2016, 2016, pp. 499–515. doi:10.1007/978-3-319-46478-7_31.

[68] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 12993–13000.

[69] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, M. Lagrange, M. D. Plumbley, A Database and Challenge for Acoustic Scene Classification and Event Detection, in: 21st European Signal Processing Conference (EUSIPCO 2013), 2013, pp. 1–5.

[70] A. Mesaros, T. Heittola, T. Virtanen, Metrics for Polyphonic Sound Event Detection, Applied Sciences 6 (6) (2016). doi:10.3390/app6060162.

[71] B. Frenay, M. Verleysen, Classification in the Presence of Label Noise: A Survey, IEEE Transactions on Neural Networks and Learning Systems 25 (5) (2014) 845–869. doi:10.1109/TNNLS.2013.2292894.

[72] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, X. Serra, Learning Sound Event Classifiers from Web Audio with Noisy Labels, in: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 21–25. doi:10.1109/ICASSP.2019.8683158.

[73] M. Harju, I. M. Morato, A. Mesaros, Does Paid Crowdsourcing Still Pay Off? Sifting Through Annotator Noise in Crowdsourced Audio Labels, in: Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE), 2024, pp. 56–60.

[74] C. Castorena, M. Cobos, F. J. Ferri, Mejoras en la Detección de Eventos Acústicos Mediante la Ponderación de Désequilibrio de Actividad y Predicciones Débiles Basadas en el Máximo, in: Tecniacustica, 2023, pp. –.

[75] V. Arora, M. Sun, C. Wang, Deep Embeddings for Rare Audio Event Detection with Imbalanced Data, in: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3297–3301. doi:10.1109/ICASSP.2019.8682395.

[76] N. Turpault, R. Serizel, Training Sound Event Detection On A Heterogeneous Dataset (2020). doi:10.48550/arXiv.2007.03931.

[77] X. Zheng, Y. Song, I. McLoughlin, L. Liu, L.-R. Dai, An Improved Mean Teacher Based Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 356–360. doi:10.1109/ICASSP39728.2021.9414931.

[78] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, S. Misra, Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review, Electronics 11 (22) (2022). doi:10.3390/electronics11223795.

[79] J. Abeßer, M. Müller, Towards Audio Domain Adaptation for Acoustic Scene Classification using Disentanglement Learning (2021). doi:10.48550/arXiv.2110.13586.

[80] S. Han, H. Mao, W. J. Dally, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, in: 4th International Conference on Learning Representations (ICLR 2016), 2016, pp. –.

[81] A. Kuzmin, M. Nagel, M. van Baalen, A. Behboodi, T. Blankevoort, Pruning vs Quantization: Which is Better?, in: Conference on Neural Information Processing Systems (NeurIPS 2023), Vol. 36, 2023, pp. 62414–62427. doi:10.48550/arXiv.2307.02973.

[82] B. Rokh, A. Azarpeyvand, A. Khanteymoori, A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification, ACM Transactions on Intelligent Systems and Technology 14 (2023). doi:10.1145/3623402.

[83] H. Ahna, T. Chen, N. Alnaasan, A. Shafi, M. Abduljabbar, H. Subramoni, D. K. Panda, Performance Characterization of Using Quantization for DNN Inference on Edge Devices, in: 7th International Conference on Fog and Edge Computing (ICFEC), 2023, pp. 1–6. doi:10.1109/ICFEC57925.2023.00009.

[84] J. Bai, F. Lu, K. Zhang, ONNX: Open Neural Network Exchange GitHub (2023).
URL https://github.com/onnx/onnx

[85] W. Li, K. Gkritza, C. Albrecht, The Culture of Distracted Driving: Evidence from a Public Opinion Survey in Iowa, Transportation Research Part F: Traffic Psychology and Behaviour 26 (2014) 337–347. doi:10.1016/j.trf.2014.01.002.

[86] F. Prat, M. Planes, M. Gras, M. Sullman, An Observational Study of Driving Distractions on Urban Roads in Spain, Accident Analysis & Prevention 74 (2015) 8–16. doi:10.1016/j.aap.2014.10.003.

[87] F. Prat, M. Gras, M. Planes, S. Font-Mayolas, M. Sullman, Driving Distractions: An Insight Gained from Roadside Interviews on Their Prevalence and Factors Associated with Driver Distraction, Transportation Research Part F: Traffic Psychology and Behaviour 45 (2017) 194–207. doi:10.1016/j.trf.2016.12.001.

[88] C. M. Farmer, K. A. Braitman, A. K. Lund, Cell Phone Use While Driving and Attributable Crash Risk, Traffic Injury Prevention 11 (5) (2010) 466–470. doi:10.1080/15389588.2010.494191.

[89] W. H. Organization, Global Status Report on Road Safety 2018: Summary (No. WHO/NMH/NVI/18.20), World Health Organization (2018).

[90] C. Stothart, A. Mitchum, C. Yehnert, The Attentional Cost of Receiving a Cell Phone Notification, Journal of Experimental Psychology: Human Perception and Performance 41 (4) (2015) 893–897. doi:10.1037/xhp0000100.

[91] J. J. Bernstein, J. Bernstein, Texting at the Light and Other Forms of Device Distraction Behind the Wheel, BMC Public Health 15 (2015) 1–5. doi:10.1186/s12889-015-2343-8.

[92] S. Edwards, L. Wundersitz, et al., Distracted Driving: Prevalence and Motivations, Accident Analysis & Prevention 54 (2019) 99–107.

[93] C. Huisingh, R. Griffin, G. McGwin Jr, The Prevalence of Distraction Among Passenger Vehicle Drivers: A Roadside Observational Approach, Traffic Injury Prevention 16 (2) (2015) 140–146.

[94] M. J. Sullman, F. Prat, D. K. Tasci, A Roadside Study of Observable Driver Distractions, Traffic Injury Prevention 16 (6) (2015) 552–557. doi:10.1080/15389588.2014.989319.

[95] S. M. Simmons, J. K. Caird, P. Steel, A Meta-Analysis of In-Vehicle and Nomadic Voice-Recognition System Interaction and Driving Performance, Accident Analysis & Prevention 106 (2017) 31–43. doi:10.1016/j.aap.2017.05.013.

[96] J. D. Lee, B. Caven, S. Haake, T. L. Brown, Speech-Based Interaction with In-Vehicle Computers: The Effect of Speech-Based E-Mail on Drivers Attention to the Roadway, Human Factors 43 (4) (2001) 631–640. doi:10.1518/001872001775870340.

[97] F. Mapfre, Teléfono Móvil, Cansancio, Somnolencia y Distracciones al Volante, https://www.fundacionmapfre.org/publicaciones/todas/movil-cansancio-somnolencia-distracciones-volante/ (2021).

[98] S. Koppel, J. Charlton, C. Kopinathan, D. Taranto, Are Child Occupants a Significant Source of Driving Distraction?, Accident Analysis & Prevention 43 (3) (2011) 1236–1244. doi:10.1016/j.aap.2011.01.005.

[99] U. Gazder, K. J. Assi, Determining Driver Perceptions About Distractions and Modeling Their Effects on Driving Behavior at Different Age Groups, Journal of Traffic and Transportation Engineering (English Edition) 9 (1) (2022) 33–43. doi:10.1016/j.jtte.2020.12.005.

[100] M. A. Regan, O. Oviedo-Trespalacios, Driver Distraction: Mechanisms, Evidence, Prevention, and Mitigation, in: The Vision Zero Handbook: Theory, Technology and Management for a Zero Casualty Policy, Springer, 2022, pp. 1–62.

[101] Y. Ma, V. Sanchez, S. Nikan, D. Upadhyay, B. Atote, T. Guha, Real-Time Driver Monitoring Systems Through Modality and View Analysis, arXiv preprint arXiv:2210.09441 (2022). doi:10.48550/arXiv.2210.09441.

[102] G. Tzanetakis, P. Cook, Musical Genre Classification of Audio Signals, IEEE Transactions on Speech and Audio Processing 10 (5) (2002) 293–302. doi:10.1109/TSA.2002.800560.

[103] J. Salamon, C. Jacoby, J. P. Bello, A Dataset and Taxonomy for Urban Sound Research, in: Proceedings of the 22nd ACM International Conference on Multimedia, Association for Computing Machinery, 2014, pp. 1041–1044. doi:10.1145/2647868.2655045.

[104] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, M. Ritter, Audio Set: An Ontology and Human-Labeled Dataset for Audio Events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 776–780. doi:10.1109/ICASSP.2017.7952261.

[105] N. Torosyan, Old Phone Ringtones as MIDI (Oct 2019) [cited 2023].
URL https://www.kaggle.com/datasets/narektorosyan/old-phone-ringtones-as-midi

[106] ASR-CabNois: A Cabin Noise Dataset (2023) [cited 2023].
URL https://magichub.com/datasets/in-vehicle-noise-dataset/

[107] Freesound (2023) [cited 2023].
URL https://freesound.org/home/

[108] J. Wilkins, P. Seetharaman, A. Wahl, B. Pardo, VocalSet: A Singing Voice Dataset (2018). doi:10.5281/zenodo.1193957.

[109] R. Serizel, N. Turpault, A. Shah, J. Salamon, Sound Event Detection in Synthetic Domestic Environments, in: ICASSP 2020 - 45th International Conference on Acoustics, Speech, and Signal Processing, Barcelona, Spain, 2020, pp. 86–90.
URL 10.1109/ICASSP40776.2020.9054478

[110] Valani, Donate-a-Cry Corpus Features Dataset (2023) [cited 2023].
URL https://www.kaggle.com/datasets/bhoomikavalani/donateacrycorpusfeaturesdataset

[111] J. Salamon, D. MacConnell, M. Cartwright, P. Li, J. P. Bello, Scaper: A Library for Soundscape Synthesis and Augmentation, in: 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), IEEE, 2017, pp. 344–348. doi:10.1109/WASPAA.2017.8170052.

[112] F. van Saane, Impulse Responses (2004) [cited 2023].
URL https://fokkie.home.xs4all.nl/IR.htm

[113] F. Burkhardt, W. F. Sendlmeier, F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, B. Weiss, A Database of German Emotional Speech, in: Proc. Interspeech 2005, 2005, pp. 1517–1520. doi:10.21437/Interspeech.2005-446.

[114] M. M. Duville, L. M. Alonso-Valerdi, D. I. Ibarra-Zarate, Mexican Emotional Speech Database Based on Semantic, Frequency, Familiarity, Concreteness, and Cultural Shaping of Affective Prosody, Data 6 (2021) 130. doi:10.3390/DATA6120130.

[115] S. R. Livingstone, F. A. Russo, The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English, PLOS ONE 13 (2018) e0196391. doi:10.1371/JOURNAL.PONE.0196391.

[116] K. Zhou, B. Sisman, R. Liu, H. Li, Seen and Unseen Emotional Style Transfer for Voice Conversion with a New Emotional Speech Dataset, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2021-June (2021) 920–924. doi:10.1109/ICASSP39728.2021.9413391.

[117] C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, S. S. Narayanan, IEMOCAP: Interactive Emotional Dyadic Motion Capture Database, Language Resources and Evaluation 42 (2008) 335–359. doi:10.1007/S10579-008-9076-6/FIGURES/10.

[118] R. Lotfian, C. Busso, Building Naturalistic Emotionally Balanced Speech Corpus by Retrieving Emotional Speech from Existing Podcast Recordings, IEEE Transactions on Affective Computing 10 (2019) 471–483. doi:10.1109/TAFFC.2017.2736999.

[119] R. Serizel, N. Turpault, H. Eghbal-Zadeh, A. P. Shah, Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments, in: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), 2018, pp. 19–23. doi:10.48550/arXiv.1807.10501.

[120] IEEE-AASP, DCASE 2023 challenge website (2023) [cited 2023].
URL https://dcase.community/challenge2023/index

[121] L. Xu, L. Wang, S. Bi, H. Liu, J. Wang, Semi-Supervised Sound Event Detection with Pre-Trained Model, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10095687.

[122] N. Shao, X. Li, X. Li, Fine-Tune the Pretrained ATST Model for Sound Event Detection, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024, pp. 911–915. doi:10.1109/ICASSP48485.2024.10446159.

[123] Y. Gong, Y. A. Chung, J. Glass, Ast: Audio Spectrogram Transformer, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2021, pp. –. doi:10.21437/Interspeech.2021-698.

[124] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, F. Wei, Beats: Audio Pre-Training with Acoustic Tokenizers, arXiv preprint arXiv:2212.09058 (2022).

[125] F. Ronchini, R. Serizel, N. Turpault, S. Cornell, The Impact of Non-Target Events in Synthetic Soundscapes for Sound Event Detection, in: Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021), Barcelona, Spain, 2021, pp. 115–119. doi:10.5281/zenodo.5770113.

[126] S. Venkatesh, D. Moffat, E. R. Miranda, You Only Hear Once: a YOLO-like Algorithm for Audio Segmentation and Sound Event Detection, Applied Sciences 12 (2022) 3293.

[127] S. Tiwari, K. Lakhotia, M. Mulimani, Evaluating Robustness of You Only Hear Once (YOHO) Algorithm on Noisy Audios in the VOICe Dataset, in: 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2021, pp. –.

[128] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural

Networks for Mobile Vision Applications (2017). doi:10.48550/arXiv.1704.04861.

[129] S. Gharib, K. Drossos, E. Fagerlund, T. Virtanen, VOICe Dataset (Jan. 2020). doi:10.5281/zenodo.3514950.

[130] G. Jocher, YOLOv5 sota realtime instance segmentation (May 2020). doi:10.5281/zenodo.3908559.
URL https://github.com/ultralytics/{YOLO}v5

[131] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal Speed and Accuracy of Object Detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 163–178.

[132] Z. Huang, M. Dong, Q. Mao, Y. Zhan, Speech Emotion Recognition Using CNN, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 801–804. doi:10.1145/2647868.2654984.

[133] R. Begazo, A. Aguilera, I. Dongo, Y. Cardinale, A Combined CNN Architecture for Speech Emotion Recognition, Sensors 24 (17) (2024). doi:10.3390/s24175797.

[134] S. Mishra, N. Bhatnagar, P. P, S. T. R, Speech Emotion Recognition and Classification Using Hybrid Deep CNN and BiLSTM Model, Multimedia Tools and Applications 83 (13) (2024) 37603–37620. doi:10.1007/s11042-023-16849-x.

[135] Y. Liu, A. Chen, G. Zhou, J. Yi, J. Xiang, Y. Wang, Combined CNN LSTM with Attention for Speech Emotion Recognition Based on Feature-Level Fusion, Multimedia Tools and Applications 83 (21) (2024) 59839–59859. doi:10.1007/s11042-023-17829-x.

[136] X. Tang, J. Huang, Y. Lin, T. Dang, J. Cheng, Speech Emotion Recognition via CNN-Transformer and Multidimensional Attention Mechanism, Speech Communication 171 (2025) 103242. doi:10.1016/j.specom.2025.103242.

[137] H. S. Kumbhar, S. U. Bhandari, Speech Emotion Recognition using MFCC features and LSTM network, in: 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 2019, pp. 1–3. doi:10.1109/ICCUBEA47591.2019.9129067.

[138] S. Zhang, X. Zhao, Q. Tian, Spontaneous Speech Emotion Recognition Using Multiscale Deep Convolutional LSTM, IEEE Transactions on Affective Computing 13 (2) (2022) 680–688. doi:10.1109/TAFFC.2019.2947464.

[139] Z. Zhu, W. Dai, Y. Hu, J. Li, Speech Emotion Recognition Model Based on Bi-GRU and Focal Loss, Pattern Recognition Letters 140 (2020) 358–365. doi:10.1016/j.patrec.2020.11.009.

[140] J. A. Russell, A Circumplex Model of Affect, Journal of Personality and Social Psychology 39 (6) (1980) 1161–1178. doi:10.1037/h0077714.

[141] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, A. Alamri, Audio–Visual Emotion-Aware Cloud Gaming Framework, IEEE Transactions on Circuits and Systems for Video Technology 25 (12) (2015) 2105–2118. doi:10.1109/TCSVT.2015.2444731.

[142] K.-J. Oh, D. Lee, B. Ko, H.-J. Choi, A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation, in: Proceedings of the 2017 18th IEEE International Conference on Mobile Data Management (MDM), 2017, pp. 371–375. doi:10.1109/MDM.2017.64.

[143] S. Mekruksavanich, A. Jitpattanakul, N. Hnoohom, Negative Emotion Recognition Using Deep Learning for Thai Language, in: Proceedings of the 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 2020, pp. 71–74. doi:10.1109/ECTIDAMTNCON48261.2020.9090768.

[144] Y. Yang, Y. Zhang, Z. Zhong, W. Dai, Y. Chen, M. Chen, Intelligent In-Car Emotion Regulation Interaction System Based on Speech Emotion Recognition, in: Proceedings of the 2024 4th International Conference on Computer, Control and Robotics (ICCCR), 2024, pp. 142–150. doi:10.1109/ICCCR61138.2024.10585371.

[145] M. Abdelwahab, C. Busso, Incremental Adaptation Using Active Learning for Acoustic Emotion Recognition, in: Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 5160–5164. doi:10.1109/ICASSP.2017.7953140.

[146] C. Lu, Y. Zong, W. Zheng, Y. Li, C. Tang, B. W. Schuller, Domain Invariant Feature Learning for Speaker-Independent Speech Emotion Recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 30 (2022) 2217–2230. doi:10.1109/TASLP.2022.3178232.

[147] M. Abdelwahab, C. Busso, Domain Adversarial for Acoustic Emotion Recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 26 (12) (2018) 2423–2435. doi:10.1109/TASLP.2018.2867099.

[148] J.-B. Kim, J.-S. Park, Y.-H. Oh, Speaker-Characterized Emotion Recognition using Online and Iterative Speaker Adaptation, Cognitive Computation 4 (4) (2012) 398–408. doi:10.1007/s12559-012-9132-9.
URL https://doi.org/10.1007/s12559-012-9132-9

[149] J.-B. Kim, J.-S. Park, Multistage Data Selection-Based Unsupervised Speaker Adaptation for Personalized Speech Emotion Recognition, Engineering Applications of Artificial Intelligence 52 (2016) 126–134. doi:https://doi.org/10.1016/j.engappai.2016.02.018.

[150] M. Abdelwahab, C. Busso, Supervised Domain Adaptation for Emotion Recognition from Speech, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5058–5062. doi:10.1109/ICASSP.2015.7178934.

[151] M. Ay, L. Özbakır, S. Kulluk, B. Gülmez, G. Öztürk, S. Özer, FC-KMeans: Fixed-Centered K-Means Algorithm, Expert Systems with Applications 211 (2023) 118656. doi:10.1016/j.eswa.2022.118656.

Appendix

This appendix provides complementary analyses and additional experimental results that support and extend the main contributions presented in this thesis. These results, while not central to the core narrative, offer valuable insights into specific design choices, model behaviors, and the validity of certain assumptions.

The appendix is structured as follows:

- **Appendix A – Impact of Data Augmentation Strategies in YOLO-based Audio Detection:** A comparative analysis of augmentation techniques (Mixup, Mosaic(2), Mosaic(4)) applied to spectrogram inputs in the context of YOLOv5n training. The results demonstrate the benefits of incorporating augmentation techniques originally designed for image data into the audio domain.
- **Appendix B – Evaluation of YOLO Architecture Versions:** A comparison among different YOLO versions (YOLOv5, YOLOv8, YOLOv11) for audio event detection. This analysis justifies the choice of YOLOv5 as the foundation for this thesis by examining detection accuracy, confusion patterns, and implementation feasibility.
- **Appendix C – Incremental Adaptation Performance for All Speakers under Balanced and Unbalanced Conditions:** This section presents the results of ISA, Random and ProtoDash methods across all speakers, evaluated under both balanced and unbalanced data conditions.
- **Appendix D – Ideal Scenario with Random Sample Selection:** A reference study that explores the accuracy variations when random subsets are selected in incremental adaptation steps. This idealized scenario assumes full access to labeled data and helps highlight the impact of selective labeling strategies on model performance.
- **Appendix E – Extended Abstract in Spanish:** In compliance with institutional requirements, this appendix includes a detailed abstract in Spanish, summarizing the key contributions and findings of this thesis.

A Impact of Data Augmentation Strategies in YOLO-based Audio Detection

The impact of various data augmentation techniques on the training of a YOLO-based model was evaluated using the YOLOv5n architecture. The techniques considered include Mixup and Mosaic, with the latter tested under two configurations: combining either 2 (Mosaic(2)) or 4 (Mosaic(4)) spectrograms. A baseline without augmentation was also included for reference.

Table 11 presents the performance metrics obtained under each augmentation setting. The results show that models trained without augmentation achieve the lowest values across all metrics. Mixup leads to a clear improvement, especially in AUC, F1, and PSDS1, while both configurations of Mosaic consistently outperform the other strategies across all evaluated metrics. The difference between Mosaic(2) and Mosaic(4) is relatively small, but the latter achieves slightly higher scores, particularly in PSDS1 and AUC, suggesting a mild benefit from incorporating a greater number of spectrograms per image.

Figure 43 provides a more detailed view of the F1 score as a function of the detection threshold θ. While Mixup achieves a high peak F1 value, this occurs within a narrow threshold range, and its performance declines sharply outside that window. This behavior suggests that Mixup leads to models that are more sensitive to decision boundary shifts and potentially less robust in real-world scenarios where the optimal threshold may vary. In contrast, both Mosaic(2) and Mosaic(4) produce smoother F1 curves, maintaining high performance across a broader range of thresholds. This indicates greater stability and a better balance between precision and recall under varying operating conditions.

Although Mosaic augmentation is a well-established technique in computer vision, particularly in object detection tasks using YOLO-based architectures, its application to audio tasks is virtually nonexistent. This is especially understandable given the limited intuitive sense of stacking multiple Mel

Table 11: Comparison of augmentation techniques on YOLOv5n performance metrics.

	None	Mixup	Mosaic(2)	Mosaic(4)
PSDS1	0.00	0.30	0.39	**0.46**
PSDS0	0.01	0.39	0.52	**0.56**
F1(1)	0.24	0.47	0.56	**0.60**
F1(0)	0.28	0.55	0.64	**0.68**
AUC	0.19	0.74	0.81	**0.85**
$AUC_{0.1}$	0.17	0.61	0.74	**0.78**
F1	0.24	0.59	0.72	**0.75**
ACC	0.91	0.93	**0.96**	**0.96**
G-mean	0.43	0.78	0.82	**0.84**

spectrograms along the frequency axis or rearranging their positions vertically within the canvas. However, the experiments presented here demonstrate that such augmentation techniques are indeed necessary, as they force the model to learn more complex time-frequency patterns by varying the relative positions of audio events. Additionally, in the case of YOLO, Mosaic enables the introduction of a larger number of samples simultaneously, which further benefits the model's ability to generalize.

B Evaluation of YOLO Architecture Versions

The selection of the base YOLO model version is a key decision that affects both current performance and the ease of future modifications and optimizations. To this end, three recent variants—YOLOv5, YOLOv8, and YOLOv11—were evaluated. The comparison considered not only performance metrics but also the simplicity and efficiency of each model, which are essential for agile and sustainable iterative development.

The confusion matrices shown in Figure 44 were generated using a simple intersection criterion between detected events and labels. While this approach is not the most common in sound event detection tasks, it defines a TP when the

temporal overlap between a predicted and a real event exceeds 50%. Although this criterion may not capture all aspects of system performance, it enables fair and consistent comparisons across models. Moreover, it conceptually aligns with more modern evaluation metrics such as PSDS, which also consider partial event alignment.

Unlike standard classification problems, these confusion matrices include a background class, which represents both false positives and false negatives. This allows errors to be visualized explicitly, whether due to missed detections of real events or detections of non-existent ones.

Analysis of the results reveals that YOLOv5 and YOLOv8 exhibit similar behavior, particularly with a recurring confusion between the Notificaciones and Ringtone classes. This confusion likely stems from the acoustic similarity between these event types, leading to misclassification or undetected instances of Notificaciones. In contrast, YOLOv11 demonstrates improved detection of the Notificaciones class, but at the cost of completely losing accurate detections of Ringtone, which are not correctly identified in any case. For the remaining classes, no significant differences were observed among the models.

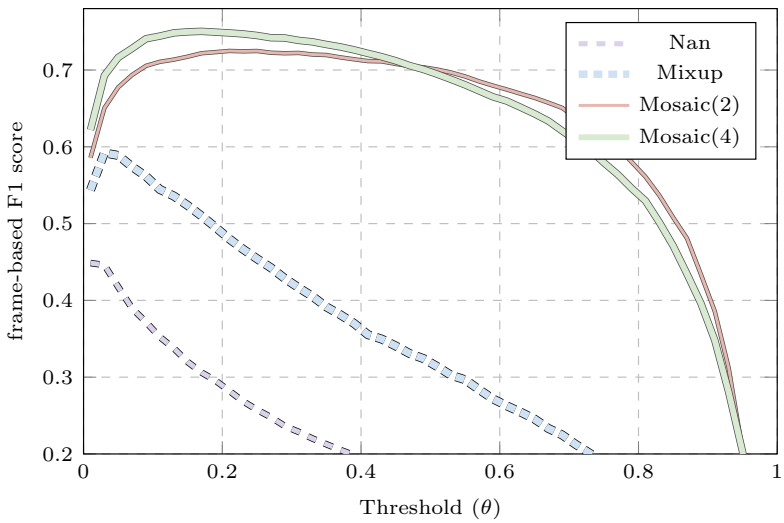
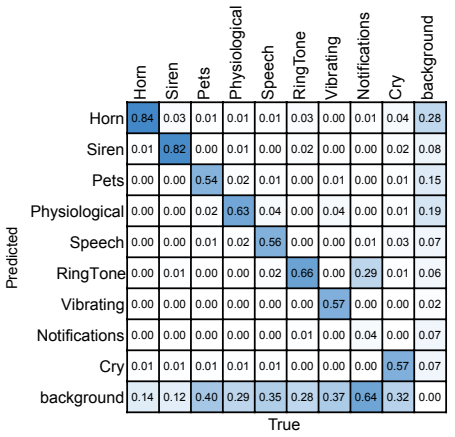
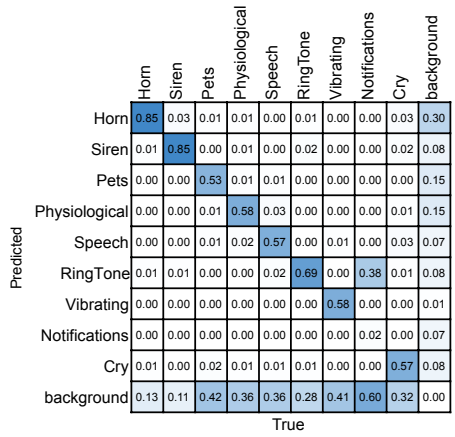


Figure 43: F1 score across detection thresholds for each augmentation strategy.

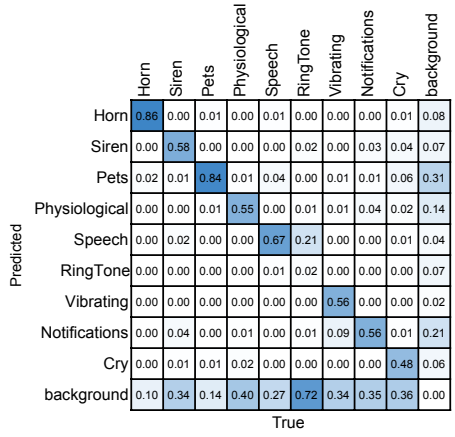
Since the overall performance differences are marginal, and considering that YOLOv5 is one of the most thoroughly documented versions with a clean and accessible codebase, it is selected as the foundation for further analysis. This choice provides a robust and competitive platform relative to newer versions, while ensuring that future development efforts benefit from a mature, well-supported, and easily modifiable framework.



(a) YOLOv5n



(b) YOLOv8n



(c) YOLOv11n

Figure 44: Confusion matrices on the synthetic evaluation set `SED_SYN_TEST` for three YOLO model variants.

C Incremental Adaptation Performance for All Speakers

This appendix presents the incremental adaptation results for all 20 target domain speakers, using the ISA Random and ProtoDash methods under both balanced and unbalanced data conditions. The speakers are indexed from 0 to 19 following the original ESD dataset setup: speakers 0 to 9 correspond to native Chinese speakers, while speakers 10 to 19 correspond to native English speakers.

The complete set of results allows for a comprehensive evaluation of the general behavior and effectiveness of the adaptation methods, where the most notable findings are discussed in Chapter 5 of the main body of the thesis.

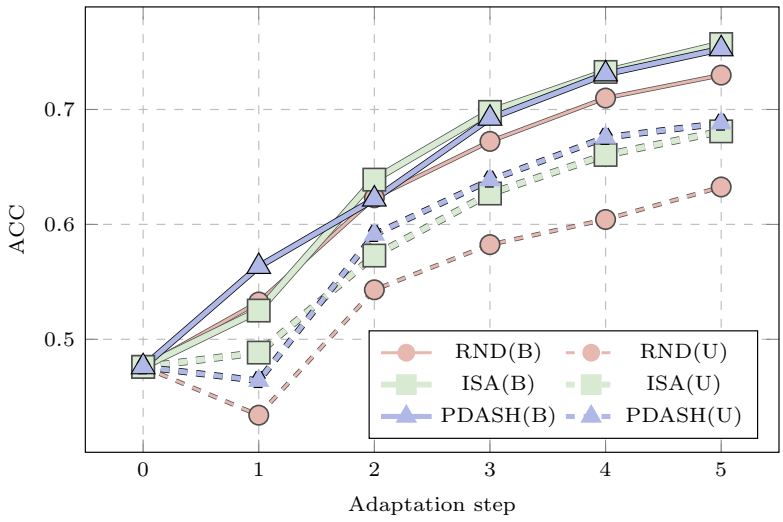


Figure 45: Incremental adaptation performance for Speaker 0 (Chinese) under balanced (B) and unbalance (U) conditions.

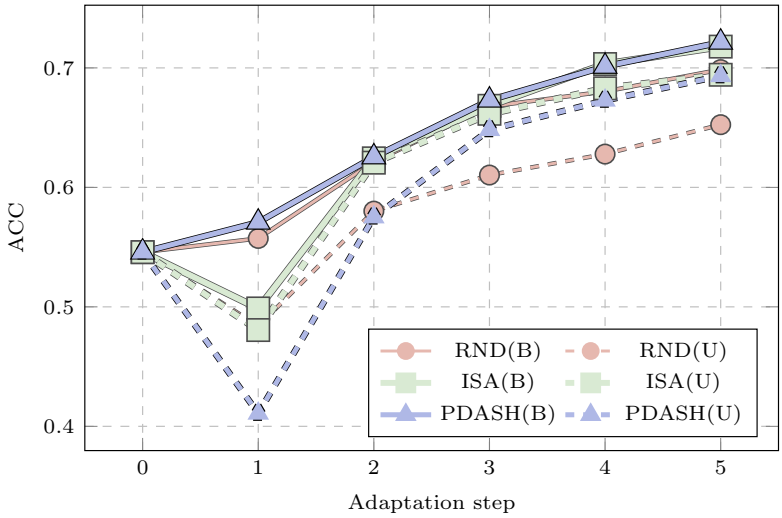


Figure 46: Incremental adaptation performance for Speaker 1 (Chinese) under balanced (B) and unbalance (U) conditions.

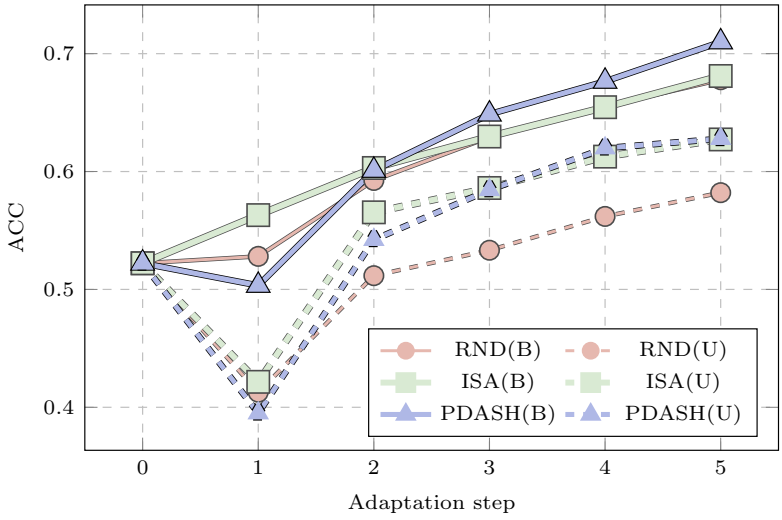


Figure 47: Incremental adaptation performance for Speaker 2 (Chinese) under balanced (B) and unbalance (U) conditions.

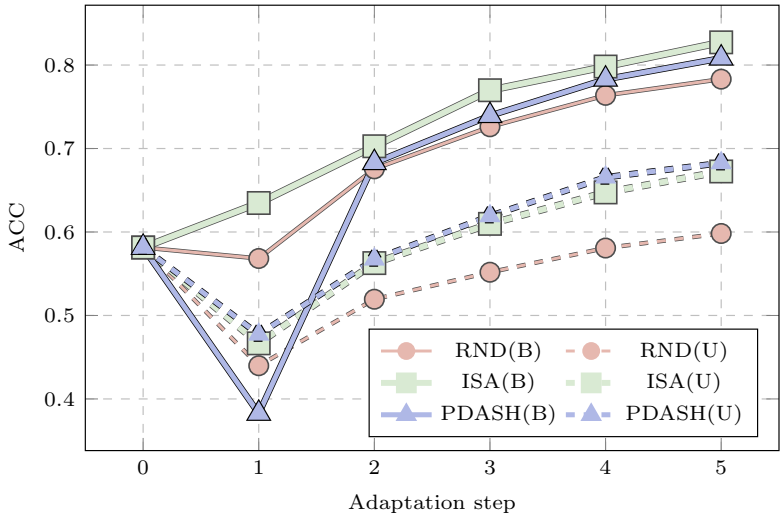


Figure 48: Incremental adaptation performance for Speaker 3 (Chinese) under balanced (B) and unbalance (U) conditions.

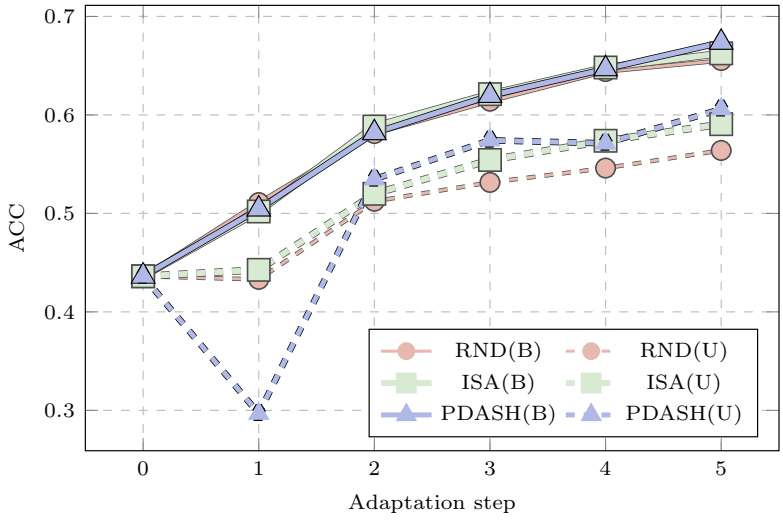


Figure 49: Incremental adaptation performance for Speaker 4 (Chinese) under balanced (B) and unbalance (U) conditions.

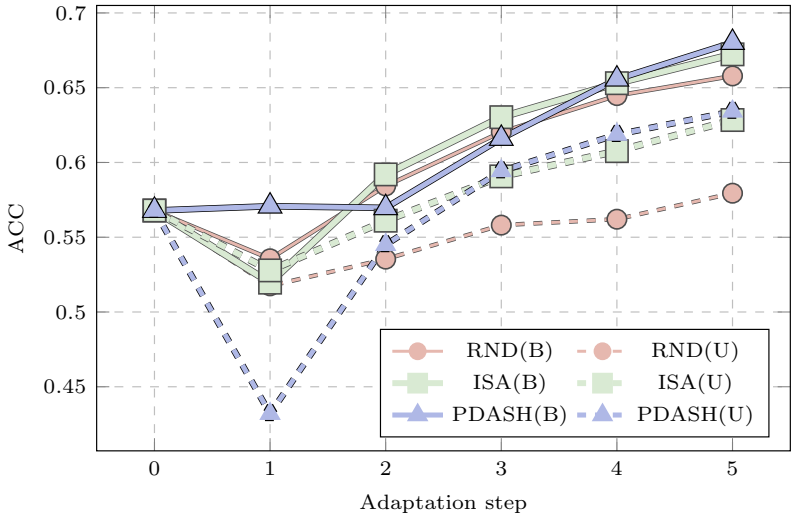


Figure 50: Incremental adaptation performance for Speaker 5 (Chinese) under balanced (B) and unbalance (U) conditions.

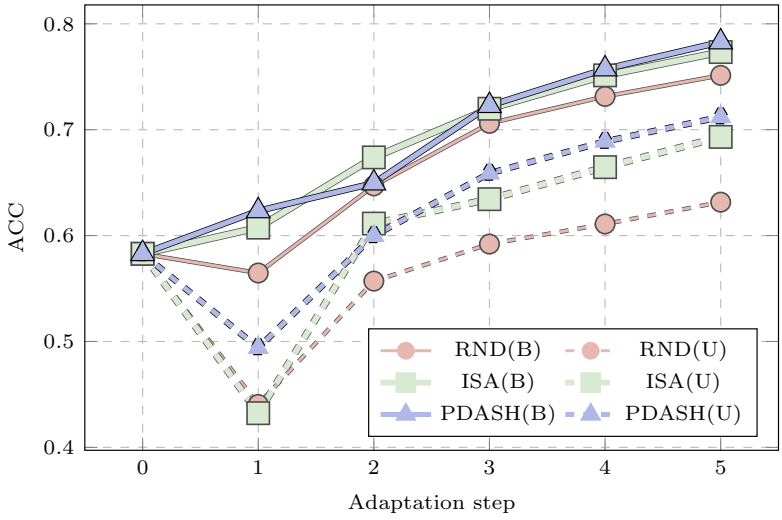


Figure 51: Incremental adaptation performance for Speaker 6 (Chinese) under balanced (B) and unbalance (U) conditions.

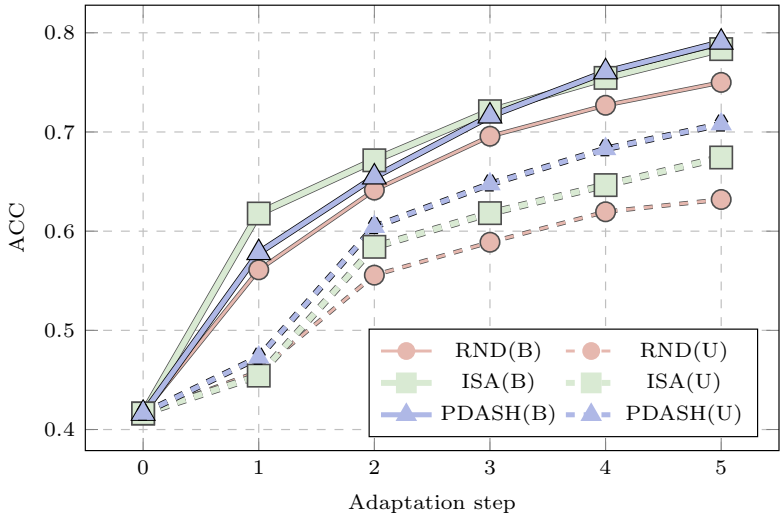


Figure 52: Incremental adaptation performance for Speaker 7 (Chinese) under balanced (B) and unbalance (U) conditions.

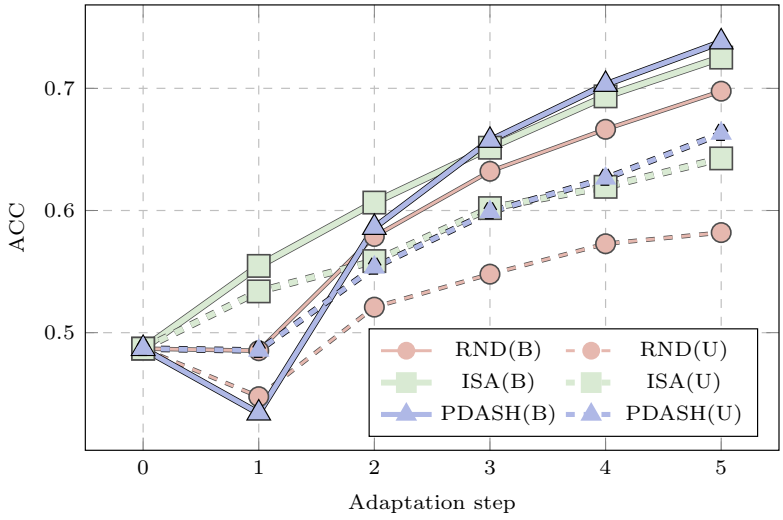


Figure 53: Incremental adaptation performance for Speaker 8 (Chinese) under balanced (B) and unbalance (U) conditions.

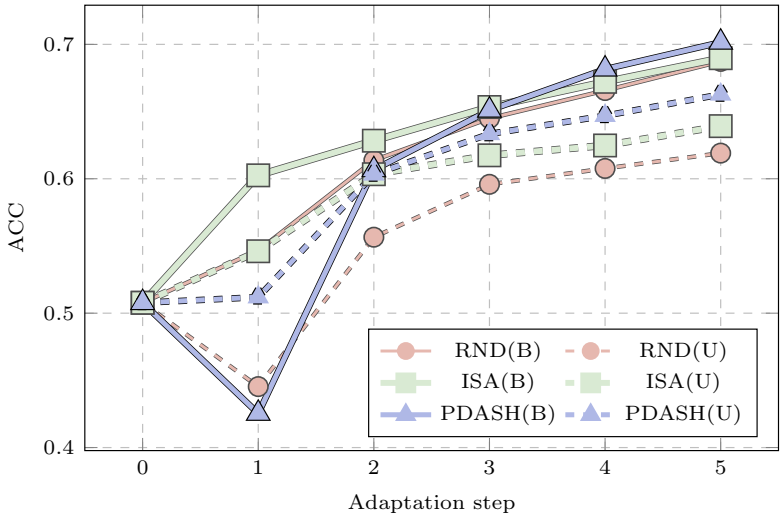


Figure 54: Incremental adaptation performance for Speaker 9 (Chinese) under balanced (B) and unbalance (U) conditions.

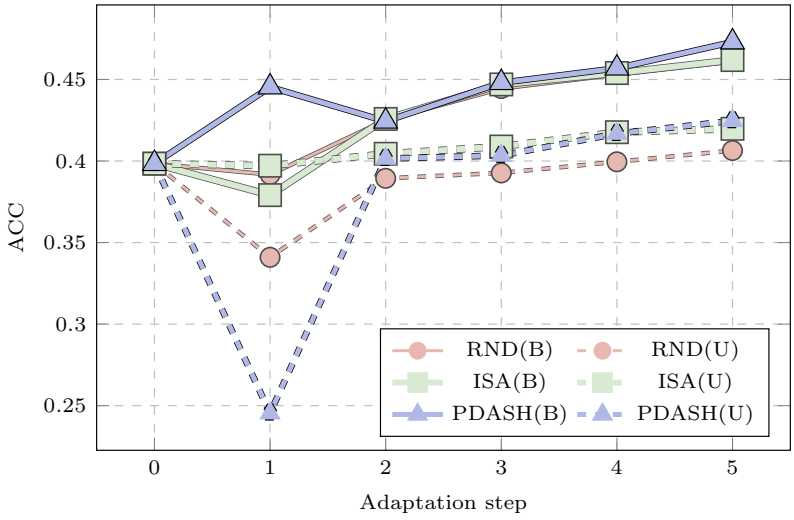


Figure 55: Incremental adaptation performance for Speaker 10 (English) under balanced (B) and unbalance (U) conditions.

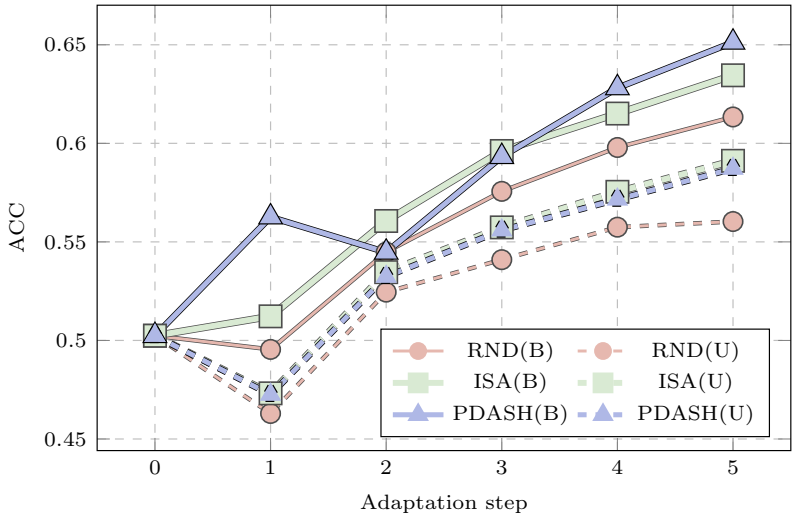


Figure 56: Incremental adaptation performance for Speaker 11 (English) under balanced (B) and unbalance (U) conditions.

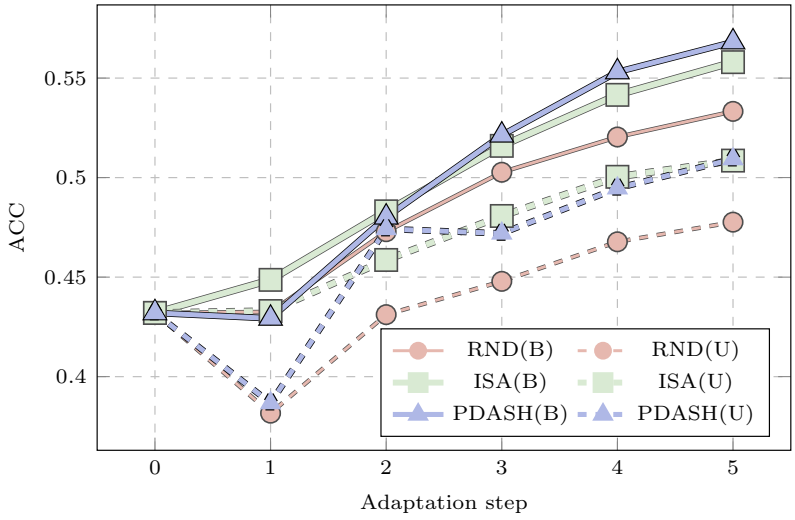


Figure 57: Incremental adaptation performance for Speaker 12 (English) under balanced (B) and unbalance (U) conditions.

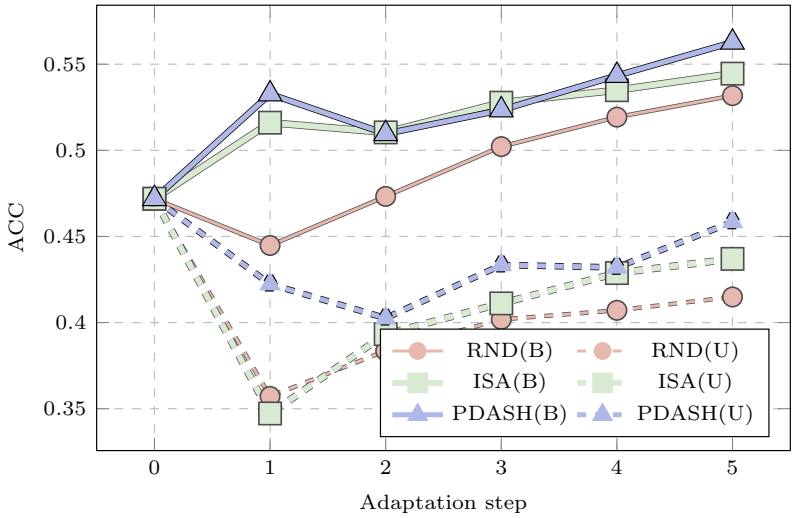


Figure 58: Incremental adaptation performance for Speaker 13 (English) under balanced (B) and unbalance (U) conditions.

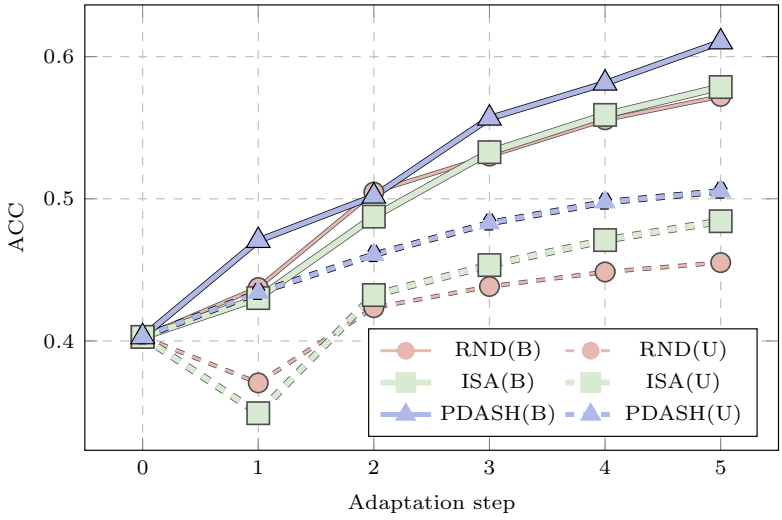


Figure 59: Incremental adaptation performance for Speaker 14 (English) under balanced (B) and unbalance (U) conditions.

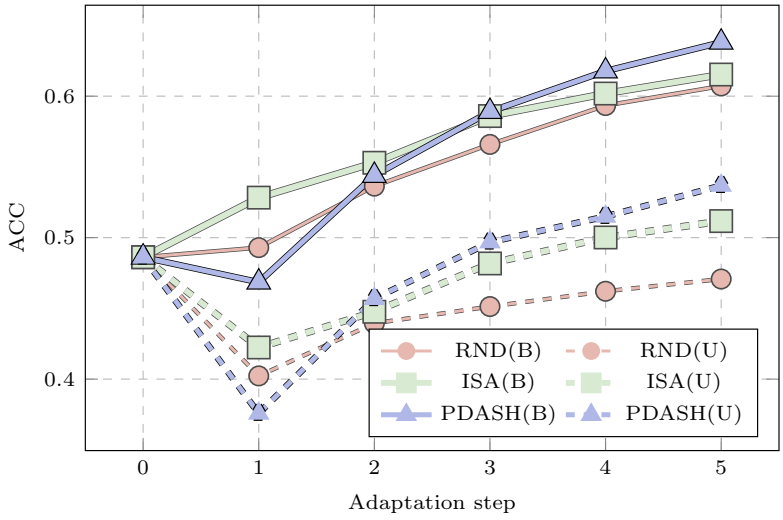


Figure 60: Incremental adaptation performance for Speaker 15 (English) under balanced (B) and unbalance (U) conditions.

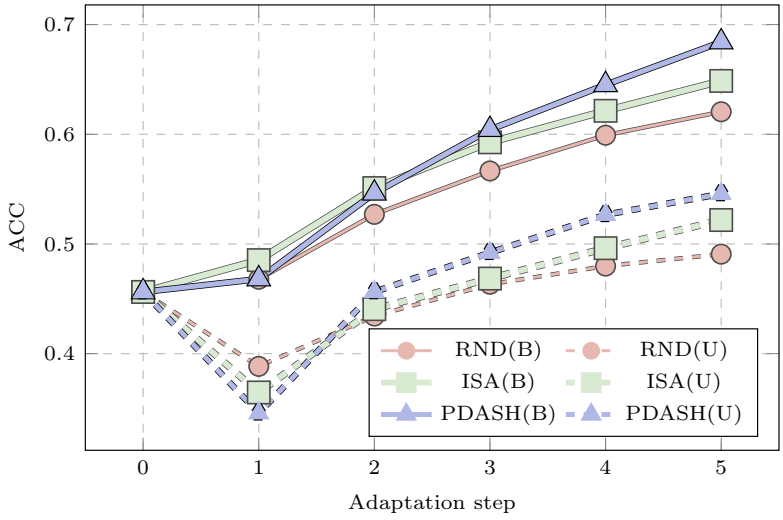


Figure 61: Incremental adaptation performance for Speaker 16 (English) under balanced (B) and unbalance (U) conditions.

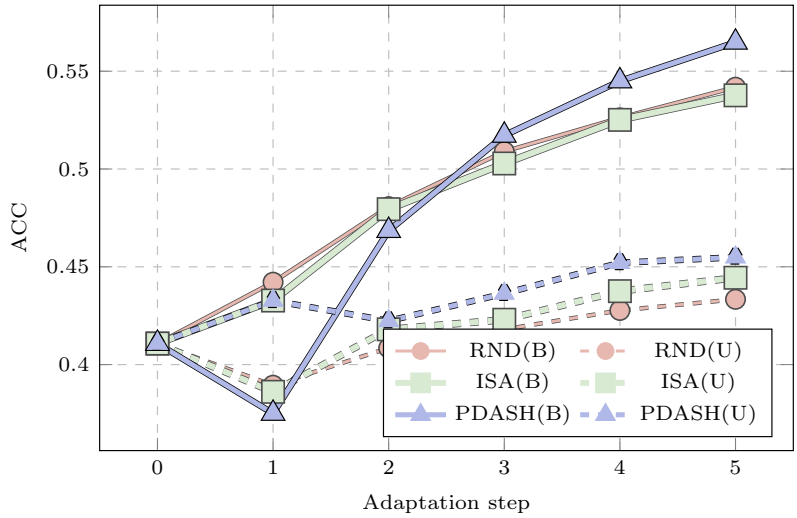


Figure 62: Incremental adaptation performance for Speaker 17 (English) under balanced (B) and unbalance (U) conditions.

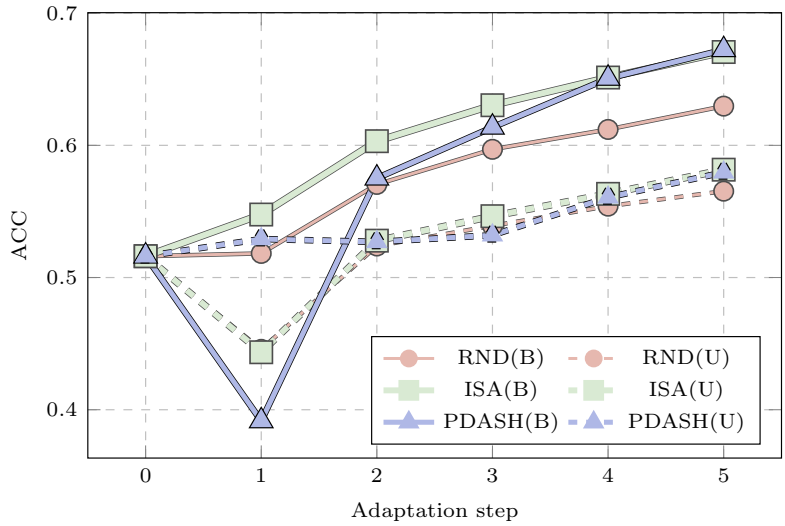


Figure 63: Incremental adaptation performance for Speaker 18 (English) under balanced (B) and unbalance (U) conditions.

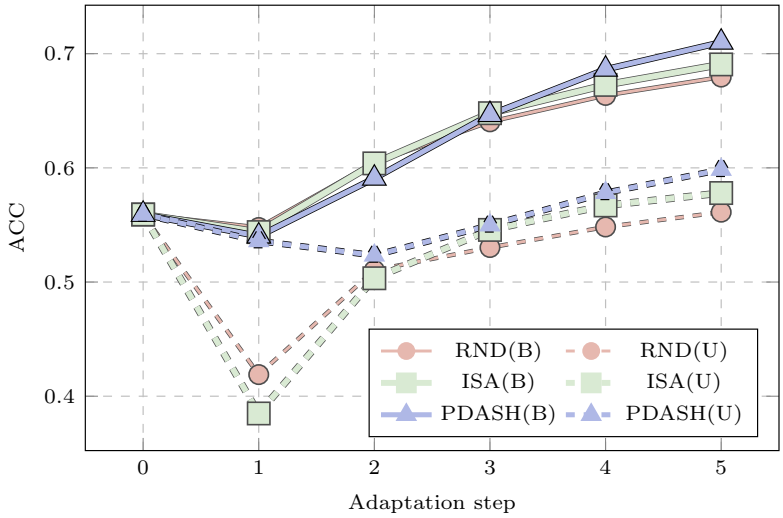


Figure 64: Incremental adaptation performance for Speaker 19 (English) under balanced (B) and unbalance (U) conditions.

D Ideal Scenario with Random Sample Selection

This reference study evaluates the baseline model performance under a random sample selection scheme using two language combinations: Chinese and English. Figures 65 show the distribution of accuracy results obtained over 200 repetitions at each incremental step. These distributions are represented as violin plots, where the dotted blue line indicates the maximum achievable accuracy at each step, the solid blue line shows the mean accuracy, and the black line reflects the performance starting from a random initial configuration. The "Maximum" line corresponds to the accuracy obtained using the full `SER_TD_TRAIN` dataset, while the "Reference" line represents the model's performance at the beginning of the adaptation process.

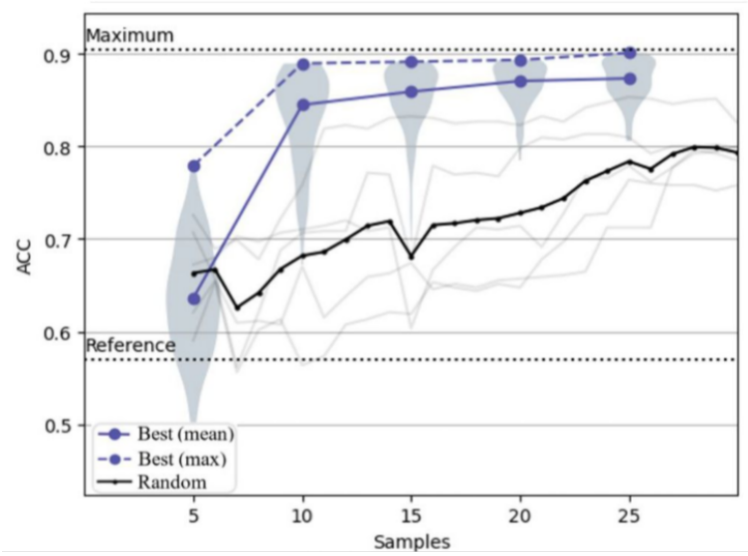
In Figure 65, for the Chinese speakers, Step 1 (5 samples) reveals a wide dispersion in accuracy scores, reflecting the variability introduced by random

sample selection. However, it is noteworthy that in some cases a selected subset yields accuracies close to 0.8 and 0.85 for each speaker, respectively. Starting from this initial state and selecting another sample set in Step 2, the performance improvement is substantial, approaching the maximum line. In contrast, starting from a random configuration results in a more modest improvement, remaining significantly below the maximum achievable value.

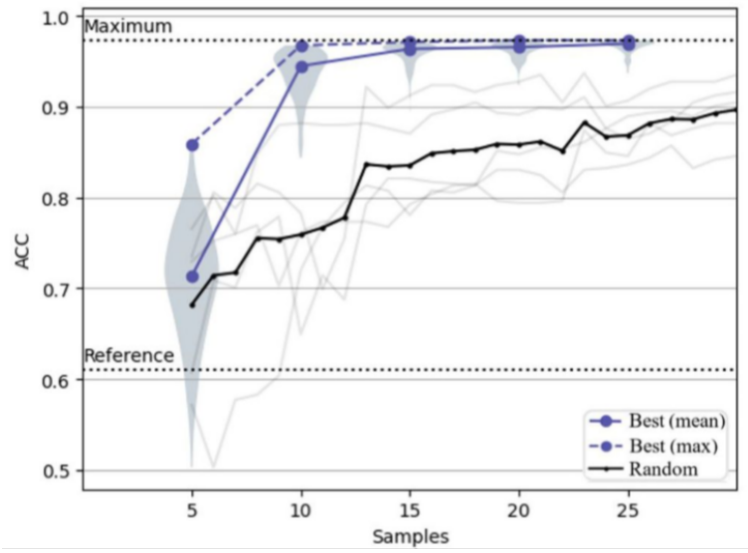
A similar trend is observed in Figure 65 for the English-speaking dataset, although maximum accuracies are lower than those for the Chinese speakers, with values around 0.60 and 0.84, respectively. This may suggest that emotion recognition in Chinese speech is somewhat easier for the model than in English.

Overall, the results indicate that a small subset of approximately 10 samples per speaker is sufficient to significantly refine the model's performance and better adapt the initial training to the target speaker. This highlights the potential of selective labeling strategies for adapting the model to new speakers, even with a limited number of samples.

It is important to note that, although this reference experiment demonstrates the potential of sample selection, it is not entirely realistic in practical scenarios. The process of generating 200 subsets, evaluating them, and labeling the best-performing subset assumes access to labels for all selected samples, which is often infeasible when labels are limited or unavailable. The purpose of this study is not to propose a practical method, but rather to illustrate the potential benefit of carefully selecting a small subset of data. By employing such strategic selection, it is possible to achieve high performance without using the entire dataset, emphasizing the importance of efficient sample selection strategies that optimize performance while minimizing the number of labeled samples required.

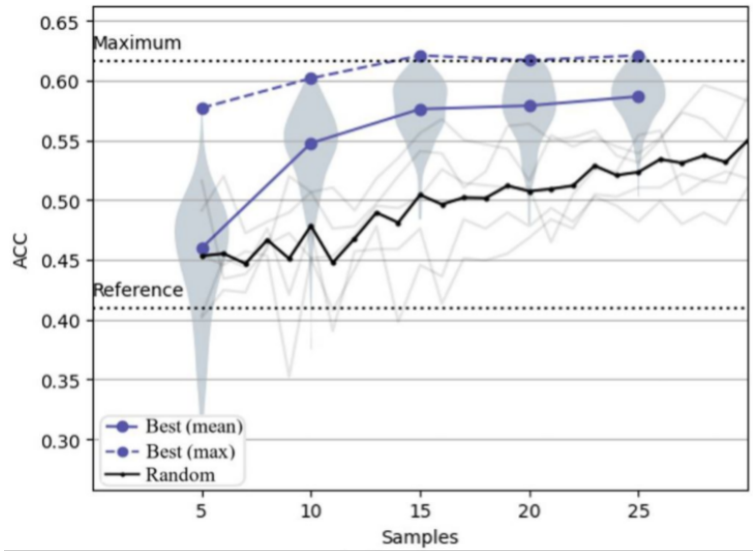


(a) Speaker 1 for `SER_TD`

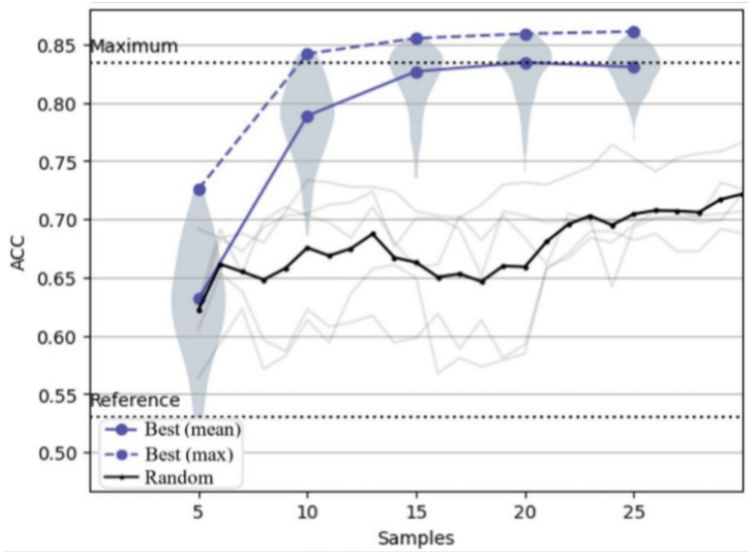


(b) Speaker 2 for `SER_TD`

Figure 65: Accuracy distributions for Chinese speakers across incremental adaptation steps.



(a) Speaker 11 for `SER_TD`



(b) Speaker 12 for `SER_TD`

Figure 66: Accuracy distributions for English speakers across incremental adaptation steps.

E Resumen extendido

Este apartado incluye un resumen extendido en español de la presente tesis doctoral, cumpliendo con los requisitos académicos establecidos por la Universitat

de València para trabajos presentados en un idioma distinto al castellano o valenciano. El objetivo de este resumen es ofrecer una visión detallada de los principales contenidos, objetivos, metodologías y contribuciones del trabajo, permitiendo su comprensión a una audiencia hispanohablante sin necesidad de recurrir al documento completo en inglés.

En los últimos años, el avance de los vehículos autónomos y los sistemas avanzados de asistencia al conductor ha concentrado la atención principalmente en la percepción visual del entorno, con tecnologías como cámaras y sensores lidar dominando el panorama. Estas herramientas han demostrado ser altamente efectivas para la detección de obstáculos, el reconocimiento de señales de tránsito y la interpretación de la dinámica vehicular en escenarios urbanos y rurales. Sin embargo, la percepción auditiva —que es igualmente esencial para la toma de decisiones seguras y eficientes— ha quedado relativamente descuidada en comparación. Esta situación resulta paradójica si consideramos que la capacidad humana para percibir sonidos fuera del campo visual —como sirenas de ambulancias, bocinas de otros vehículos o el crujido de una colisión inminente— aporta información crítica y complementaria para la seguridad vial. La incorporación de esta dimensión sensorial en sistemas inteligentes permitiría anticipar situaciones de riesgo incluso antes de que sean visibles, actuando con mayor rapidez y precisión, y proporcionando una capa adicional de conciencia situacional que puede marcar la diferencia en contextos dinámicos y complejos.

Los entornos acústicos vehiculares presentan características únicas que los diferencian de otros contextos tradicionales de procesamiento de audio. A diferencia de escenarios estáticos como estudios de grabación o espacios domésticos controlados, la cabina de un vehículo constituye un espacio cerrado con propiedades acústicas específicas, incluyendo reverberaciones y reflexiones sonoras que dependen tanto de los materiales como de la geometría interna del habitáculo. Además, se suma el aislamiento parcial de fuentes externas, que puede atenuar o distorsionar señales relevantes y hacer más difícil la identificación de sonidos importantes. El ruido de fondo, por su parte, varía de manera dinámica según múltiples factores, como la velocidad del vehículo, las condiciones del camino (asfalto, tierra, baches), el tipo de motor (eléctrico o de combustión) y el

funcionamiento de sistemas internos activos, tales como el aire acondicionado, la ventilación o los equipos de sonido. Estas particularidades hacen que la detección y clasificación de eventos sonoros en vehículos requiera el desarrollo de modelos y enfoques altamente especializados, capaces de interpretar correctamente el entorno acústico, filtrar adecuadamente la información irrelevante y mantener únicamente aquella que represente un posible riesgo para la conducción o para los ocupantes del vehículo.

Por otro lado, la movilidad intrínseca del vehículo introduce retos adicionales en términos de variabilidad temporal y espacial de las fuentes sonoras. Mientras que un sistema fijo —como el monitoreo de un espacio dentro de hogares o edificios inteligentes— puede contar con condiciones acústicas relativamente constantes y repetitivas, un sistema embebido en un vehículo debe adaptarse continuamente a distintos entornos o paisajes sonoros. Estos cambios pueden ser drásticos incluso en trayectos cortos: por ejemplo, pasar de un túnel cerrado a una autopista abierta, de una zona urbana densa a un área rural tranquila, o enfrentar condiciones meteorológicas adversas que alteren el perfil sonoro ambiental. Asimismo, debe hacer frente a cambios constantes en la posición y orientación del vehículo, variaciones en la intensidad y dirección del ruido ambiental, y a la superposición compleja de múltiples fuentes de sonido internas y externas, tanto mecánicas como humanas. Además, si se desea contribuir de manera positiva a la seguridad vial, este entendimiento acústico tiene que ser especialmente rápido y eficiente bajo condiciones de recursos limitados, ya que los sistemas embebidos en automóviles suelen contar con capacidad computacional y energética restringida. Todo lo anterior convierte el análisis de audio vehicular en una tarea compleja y multifacética, que demanda soluciones computacionales eficientes, robustas y capaces de operar en tiempo real sin comprometer el desempeño general del sistema.

En este escenario, la capacidad de un sistema vehicular para detectar y clasificar correctamente eventos sonoros relevantes puede representar una herramienta clave para mejorar la seguridad vial. La identificación temprana de señales acústicas críticas, como sirenas de emergencia, alertas de colisión, sonidos anómalos en el motor o incluso expresiones verbales de los ocupantes,

puede permitir respuestas oportunas y automáticas por parte del sistema o bien alertar al conductor sobre un riesgo potencial. Esta capacidad resulta aún más crítica cuando dichos eventos coinciden con situaciones identificadas como distractores, tales como el uso del teléfono móvil, la interacción constante entre los ocupantes del vehículo, alarmas múltiples o sonidos que desvían la atención del entorno. En estos casos, una correcta interpretación auditiva podría actuar como un mecanismo compensatorio, ayudando a recuperar o reforzar el foco del conductor ante condiciones de atención disminuida y contribuyendo así a evitar incidentes causados por distracciones.

Además, este tipo de sistemas podría ofrecer una instancia de retroalimentación posterior al recorrido, informando al conductor sobre los estímulos auditivos a los que estuvo expuesto durante el trayecto, lo cual permitiría una toma de conciencia más profunda y una mejor preparación para futuros desplazamientos. Finalmente, ante la ocurrencia de un siniestro, el análisis acústico también podría contribuir a recopilar información relevante sobre los sonidos previos al evento, ayudando a identificar posibles causas o factores de riesgo involucrados, y facilitando la reconstrucción del contexto para investigaciones posteriores.

La utilidad de un sistema de clasificación y detección de eventos sonoros distractores dentro de un entorno vehicular no depende únicamente de su capacidad para captar sonidos relevantes, sino también de su habilidad para minimizar errores, especialmente en entornos acústicamente complejos y altamente variables. La sensibilidad del sistema debe, por tanto, equilibrarse cuidadosamente con una alta precisión en la discriminación de eventos significativos, con el fin de evitar falsas alarmas que no solo resultan molestas o intrusivas, sino que también pueden disminuir la confianza del usuario en la tecnología e, incluso, convertirse paradójicamente en un nuevo distractor para el conductor. Así, se vuelve evidente la necesidad de enfoques que no solo reconozcan correctamente los sonidos críticos en tiempo real, sino que también comprendan el contexto en el que ocurren, distinguiendo de manera efectiva entre información útil y ruido inofensivo o irrelevante. Esta capacidad contextual es lo que puede marcar la diferencia entre un sistema meramente reactivo y uno

verdaderamente inteligente y proactivo en la asistencia al conductor, capaz de adaptarse a situaciones cambiantes y tomar decisiones informadas.

Con base en los desafíos expuestos y en la creciente necesidad de incorporar inteligencia auditiva en vehículos, esta tesis propone un conjunto de tres contribuciones principales que abordan el problema de la clasificación y detección de eventos sonoros distractores en entornos de conducción, mediante el uso de técnicas avanzadas de aprendizaje profundo y procesamiento de señales acústicas.

En primer lugar, se presenta el desarrollo de un marco de trabajo sistemático y riguroso orientado al análisis y modelado de eventos sonoros distractores dentro del vehículo, con especial atención a aquellos sonidos que la literatura especializada ha identificado como factores de riesgo para la atención y seguridad del conductor. Este marco incluye además la construcción y curación de un conjunto de datos específico para esta tarea, diseñado bajo condiciones controladas y situaciones reales de conducción para asegurar la representatividad y diversidad de los sonidos. En segundo lugar, se propone la implementación y optimización de modelos de clasificación y detección eficientes, capaces de operar en tiempo real y compatibles con las restricciones computacionales y energéticas propias de los sistemas embebidos en vehículos modernos. Finalmente, como tercera contribución, se exploran estrategias de adaptación incremental y aprendizaje continuo que permiten a los sistemas ajustarse dinámicamente a nuevas condiciones acústicas y escenarios no vistos previamente, aumentando así su robustez y aplicabilidad en entornos reales y cambiantes.

Como se mencionó anteriormente, un aspecto fundamental para abordar el problema de la detección de eventos sonoros distractores es entender cuáles de los sonidos presentes durante un escenario de conducción resultan verdaderamente relevantes desde el punto de vista de la seguridad vial y la atención del conductor. Con este fin, se propone una taxonomía exhaustiva de eventos acústicos vehiculares, que clasifica los sonidos más comunes y representativos según su origen y características acústicas. Esta taxonomía identifica un total de 45 eventos distintos, no todos ellos considerados distractores, pero sí útiles para representar el panorama acústico completo y variado de un entorno vehicular realista. Esta caracterización resulta especialmente valiosa al momento

de construir bases de datos sintéticas o semi-sintéticas que busquen emular condiciones de conducción reales y complejas, facilitando el entrenamiento y validación de modelos de detección.

Dentro de esta clasificación general, se destacan nueve eventos identificados como distractores, seleccionados tras contrastar nuestra propuesta con evidencia reportada en la literatura científica y técnica. Entre ellos se incluyen sonidos externos diseñados para captar la atención inmediata del conductor, como sirenas de emergencia y bocinas, los cuales, aunque necesarios para alertar sobre situaciones de riesgo, pueden provocar reacciones involuntarias que afectan la conducción segura. También se contemplan sonidos relacionados con dispositivos móviles, tales como timbres, vibraciones y notificaciones, que pueden desencadenar una pérdida de atención incluso si el conductor no interactúa directamente con el dispositivo. En el ámbito interno del vehículo, se consideran eventos acústicos producidos por los ocupantes, como la voz humana, el llanto y sonidos de mascotas, que implican una carga emocional significativa y pueden alterar la concentración. Finalmente, se incorporan sonidos fisiológicos como estornudos o toses, cuya ocurrencia, aunque breve, puede provocar distracciones momentáneas y reducir la capacidad de respuesta del conductor ante situaciones críticas.

Estos nueve eventos distractores han servido como base para la construcción de un conjunto de datos específico orientado a tareas de clasificación y detección acústica en entornos vehiculares, donde dichos sonidos se presentan tanto de forma aislada como en combinación con otros eventos acústicos comunes definidos en la taxonomía. Dado que la detección requiere anotaciones temporales precisas —es decir, marcar el inicio y el fin de cada evento sonoro—, el etiquetado manual de datos reales se vuelve una tarea compleja, intensiva en tiempo y susceptible a errores o inconsistencias. Por esta razón, el uso de datos sintéticos cobra particular relevancia: permite generar un gran volumen de muestras con etiquetas exactas y controladas, facilitando el entrenamiento supervisado de modelos con arquitecturas más complejas y exigentes. Además, la inclusión de múltiples eventos sonoros en escenarios acústicos variados y con diferentes grados de superposición mejora la capacidad de generalización del sistema frente

a condiciones reales de ruido ambiental, mezcla de fuentes y variabilidad en la intensidad o duración de los sonidos. De este modo, el dataset resultante no solo representa de forma más fiel el entorno acústico vehicular, sino que se convierte en una herramienta fundamental para el desarrollo y la evaluación de soluciones de detección robustas, escalables y realistas.

Si bien los datos sintéticos permiten un entrenamiento controlado y eficiente, la validación sobre datos reales es indispensable para garantizar la aplicabilidad y eficacia del sistema en escenarios prácticos y cotidianos. Por ello, se recopilaron grabaciones en condiciones reales de conducción, capturando la complejidad y diversidad acústica inherente a distintos trayectos, velocidades y contextos urbanos. Estos datos reales permiten evaluar el rendimiento de los modelos más allá de entornos idealizados, poniendo a prueba su capacidad de generalización, su tolerancia frente al ruido no etiquetado y su robustez ante eventos acústicos imprevistos o atípicos. Además, el uso de estos registros posibilita una comparación directa con situaciones reales donde los distractores efectivamente ocurren, ofreciendo así un marco más riguroso y representativo para la validación, ajuste fino y mejora continua del sistema.

De esta forma, tanto la taxonomía propuesta como los conjuntos de datos desarrollados constituyen los principales resultados de esta primera contribución. La taxonomía ofrece un marco conceptual claro, estructurado y replicable para clasificar eventos sonoros en contextos vehiculares, facilitando futuras investigaciones, comparaciones sistemáticas y la estandarización en el área. Por su parte, las bases de datos —una sintética, diseñada para entrenamiento intensivo y controlado, y otra real, orientada a la validación práctica y la evaluación en condiciones reales— representan recursos fundamentales para el avance del campo, al permitir el diseño, evaluación y comparación rigurosa de algoritmos en escenarios acústicos realistas y complejos. En conjunto, estos aportes establecen los cimientos necesarios para abordar de manera rigurosa, reproducible y escalable el problema de la detección y clasificación de eventos sonoros distractores en vehículos.

Sobre esta base de datos y definición conceptual, se introduce la segunda contribución principal de esta tesis: el diseño, desarrollo y evaluación de un

modelo eficiente para la detección automática de eventos sonoros en entornos vehiculares, inspirado en el enfoque de detección rápida propuesto por YOLO (You Only Look Once, por sus siglas en inglés), originalmente concebido para tareas de detección de objetos en imágenes. A diferencia de su aplicación tradicional en visión por computadora, esta versión adaptada trabaja sobre representaciones espectro-temporales del audio, procesando espectrogramas que capturan tanto la evolución temporal como la distribución frecuencial de los eventos sonoros. De este modo, el modelo no solo estima las clases correspondientes a cada evento acústico, sino que localiza con precisión sus límites temporales, identificando los puntos de inicio y fin de cada evento dentro de una señal continua y compleja.

Este enfoque permite formular la detección de eventos sonoros como un problema de regresión directa sobre los tiempos de inicio y fin de los eventos, en contraste con los métodos tradicionales basados en redes convolucionales recurrentes (CRNN), que abordan la tarea como una clasificación segmento a segmento sobre ventanas temporales. Esta diferencia metodológica es especialmente relevante en entornos acústicos complejos como el vehicular, donde los sonidos pueden solaparse parcialmente, variar considerablemente en duración o intensidad, y donde resulta más valioso detectar eventos acústicos discretos con límites temporales bien definidos, en lugar de mantener una etiqueta continua para cada cuadro de audio. En particular, el modelo basado en YOLO muestra ventajas significativas tanto en la precisión de detección como en la velocidad de inferencia, lo que lo hace especialmente adecuado para su implementación en sistemas en tiempo real, con recursos computacionales limitados y demandas estrictas de eficiencia.

A nivel experimental, las evaluaciones realizadas sobre el conjunto de datos sintético revelan mejoras significativas con respecto a los modelos de referencia (línea base). Utilizando métricas específicas para tareas de detección de eventos acústicos —como el Polyphonic Sound Detection Score (PSDS), que evalúa la calidad de las predicciones teniendo en cuenta la alineación temporal entre los inicios y finales de los eventos detectados y los eventos reales—, así como métricas clásicas basadas en ventanas temporales fijas, como el F1-score y el Área Bajo la Curva (AUC), el modelo propuesto basado en la arquitectura YOLO demuestra

una mayor capacidad para identificar correctamente eventos distractores, incluso en escenarios multiclase y de alta densidad sonora. Además, se observa una baja tasa de falsos positivos, lo que evidencia una mejor discriminación entre eventos relevantes y ruido de fondo. Esta propiedad es especialmente valiosa en entornos vehiculares, donde la saturación acústica es común y los errores de detección pueden comprometer la utilidad del sistema o generar nuevas distracciones, contrariando su propósito.

No obstante, al implementar el sistema en condiciones reales de conducción, se observa una reducción del rendimiento en comparación con los resultados obtenidos en entornos sintéticos. Esta caída, aunque esperada, pone de manifiesto los desafíos asociados a la transferencia entre dominios: la variabilidad no controlada del entorno real introduce factores como reverberaciones impredecibles, interferencias acústicas no modeladas y ruido de fondo cambiante, que afectan la precisión del sistema. A pesar de ello, un análisis más detallado de las métricas en condiciones reales muestra que el modelo conserva una utilidad práctica considerable, siendo viable como módulo de detección temprana de eventos distractores. Sin embargo, se identifican áreas críticas que requieren refinamiento, particularmente en la reducción de falsas alarmas y en la mejora de la precisión en la delimitación temporal de los eventos detectados, aspectos clave para una integración efectiva y confiable en contextos vehiculares reales.

Una de las ventajas clave del enfoque propuesto reside en su eficiencia computacional, un aspecto crítico para su adopción en sistemas vehiculares en tiempo real. Al estar compuesto exclusivamente por capas convolucionales y prescindir de módulos recurrentes —los cuales, por su naturaleza secuencial, tienden a incrementar la carga computacional—, el modelo logra una reducción sustancial en el tiempo de inferencia por muestra. Esta optimización no solo permite respuestas más ágiles ante la ocurrencia de eventos críticos, sino que también habilita el procesamiento simultáneo de múltiples flujos de audio o segmentos de mayor duración. Esta capacidad es especialmente útil para contrastar predicciones entre distintas fuentes acústicas, lo que mejora la robustez del sistema frente a capturas parciales, superposición de eventos o interferencias. En entornos embebidos con recursos limitados, como las plataformas integradas en

vehículos, esta eficiencia se traduce en mayor cobertura temporal, menor latencia de respuesta y menor consumo energético, habilitando soluciones más sostenibles, escalables y adaptables. En conjunto, estas características posicionan al modelo como una alternativa sólida y prometedora para su integración en aplicaciones reales de asistencia al conductor.

Un desafío recurrente en la implementación práctica de modelos entrenados con datos sintéticos es la degradación de su desempeño al enfrentarse a datos reales, debido a las diferencias inherentes entre ambos dominios. En este trabajo, esta discrepancia entre el dominio sintético (fuente) y el dominio real (objetivo) afecta la capacidad del modelo para generalizar de forma efectiva en escenarios de conducción cotidiana. Con el objetivo de mitigar esta brecha, la tercera contribución de esta tesis introduce una estrategia de adaptación de dominio de tipo incremental, que permite ajustar progresivamente los modelos a nuevas condiciones acústicas o incluso a usuarios específicos. Esta adaptación progresiva contribuye a cerrar la distancia entre dominios y mejora la robustez y adaptabilidad del sistema en aplicaciones reales.

La viabilidad y efectividad de esta estrategia se estudian en el contexto del reconocimiento de emociones en la voz (SER, por sus siglas en inglés: Speech Emotion Recognition), principalmente motivada por la disponibilidad de conjuntos de datos públicos adecuados y bien anotados, que permiten el diseño experimental riguroso. Esta elección, sin embargo, no es arbitraria. La detección de emociones dentro del vehículo tiene una relación directa con la seguridad vial, dado que el estado emocional del conductor puede influir significativamente en su nivel de atención, toma de decisiones y comportamiento al volante. Por tanto, aunque el estudio se centra en una tarea distinta a la detección de distractores acústicos, contribuye de forma indirecta —pero relevante— al objetivo general de esta tesis: avanzar hacia sistemas de asistencia auditiva inteligentes que mejoren la seguridad del conductor mediante una percepción más rica y adaptativa.

A diferencia de los enfoques no supervisados tradicionales para la adaptación de dominio —los cuales intentan ajustar los modelos exclusivamente a partir de datos no etiquetados del dominio objetivo—, la estrategia propuesta en esta tesis adopta un enfoque semisupervisado. Esta estrategia selecciona de

manera iterativa un pequeño subconjunto de muestras representativas dentro de un conjunto no etiquetado correspondiente a un nuevo hablante. La selección se realiza en un espacio latente generado por el propio modelo, utilizando una técnica de agrupamiento modificada diseñada para preservar los centros previamente calculados, garantizando así la representatividad de muestras en todo el espacio latente. Las muestras seleccionadas son posteriormente etiquetadas por el usuario y empleadas para actualizar el modelo, permitiendo su adaptación a las características específicas del nuevo dominio con un esfuerzo mínimo de intervención. El objetivo principal de este método es minimizar la cantidad de ejemplos etiquetados requeridos, promoviendo una adaptación práctica, eficiente y poco intrusiva para el usuario final.

Los resultados experimentales obtenidos respaldan la eficacia de esta estrategia incremental. Con la inclusión de menos de 25 muestras etiquetadas por hablante, el modelo logra mejoras significativas en la precisión de clasificación emocional, lo que demuestra el potencial del enfoque para aplicaciones en escenarios reales. Esta mejora se consigue sin necesidad de un proceso de reentrenamiento intensivo, logrando un equilibrio favorable entre el costo del etiquetado manual y el beneficio en desempeño. En el contexto vehicular, donde la personalización del sistema y su robustez frente a nuevas condiciones son aspectos clave, esta estrategia demuestra ser especialmente prometedora.

Si bien la estrategia de selección y adaptación incremental (ISA, por sus siglas en inglés) ha mostrado resultados alentadores, es importante reconocer sus limitaciones. En nuestras evaluaciones, se constató que, aunque un subconjunto cuidadosamente seleccionado puede aproximarse al rendimiento de un modelo completamente supervisado, este resultado no es garantizable para cualquier selección aleatoria. La variabilidad observada en los resultados entre distintas selecciones destaca la necesidad de contar con mecanismos de selección robustos y fundamentados. En este sentido, aunque ISA representa un avance sustancial hacia la personalización eficiente de modelos para condiciones reales —como nuevos hablantes o ambientes acústicos vehiculares cambiantes—, aún persiste una brecha respecto al rendimiento óptimo. Para superarla, será necesario desarrollar estrategias más refinadas que identifiquen de forma más efectiva las muestras

con mayor valor informativo, permitiendo así mejorar la calidad del ajuste sin incrementar significativamente el esfuerzo de anotación.

A lo largo de esta tesis se ha profundizado en el estudio del reconocimiento auditivo como una vía complementaria y robusta para mejorar la seguridad en entornos vehiculares. Las tres contribuciones principales presentadas siguen una progresión lógica: desde la identificación y caracterización de sonidos relevantes dentro del vehículo, hasta la detección eficiente en tiempo real y la adaptación personalizada de los modelos. Más allá de los avances técnicos alcanzados, los resultados obtenidos sustentan la hipótesis de que el uso combinado de aprendizaje profundo y comprensión del entorno de conducción permite desarrollar soluciones prácticas que equilibran precisión, eficiencia computacional y capacidad de adaptación. Particularmente, se ha validado que los eventos sonoros distractores no solo pueden ser identificados con alta precisión, sino también utilizados como señales clave para ajustar dinámicamente la respuesta del sistema, favoreciendo la recuperación de la atención del conductor en tiempo real.

Asimismo, esta investigación pone de relieve que el verdadero reto no radica exclusivamente en alcanzar altos niveles de rendimiento en condiciones controladas, sino en diseñar soluciones capaces de enfrentar la complejidad y variabilidad inherente al mundo real: distintas voces, diversos modelos de vehículos, condiciones acústicas cambiantes. En este contexto, la combinación de datos sintéticos, arquitecturas eficientes y estrategias de adaptación incremental se revela como un enfoque prometedor, capaz de afrontar estos desafíos sin comprometer la escalabilidad ni la usabilidad del sistema. En conjunto, las contribuciones aquí presentadas sientan las bases para el desarrollo de tecnologías auditivas inteligentes integrables en vehículos, fortaleciendo así el camino hacia sistemas de asistencia al conductor más seguros, adaptativos y sensibles al entorno real.

Las contribuciones desarrolladas en esta tesis poseen un amplio potencial de aplicación dentro del ecosistema de vehículos inteligentes, donde los sistemas de asistencia al conductor y las plataformas de monitoreo del comportamiento del usuario requieren, cada vez más, una interpretación contextual y multimodal del entorno. En este contexto, la dimensión auditiva —aún menos explotada en

comparación con la visión por computadora o los sensores físicos— representa un canal de información esencial, capaz de complementar e incluso anticipar eventos críticos para la seguridad vial.

Una aplicación inmediata consiste en integrar un sistema de detección de eventos sonoros distractores como módulo complementario dentro de los sistemas actuales de monitoreo del conductor. Estos sistemas suelen basarse en señales visuales, como la dirección de la mirada o la postura de la cabeza, pero no consideran lo que ocurre acústicamente en la cabina. La detección de eventos como conversaciones, llanto, notificaciones móviles o timbres permite inferir posibles causas de distracción, en lugar de limitarse a identificar sus manifestaciones. Esta información adicional puede alimentar motores de decisión más sofisticados, capaces de ajustar dinámicamente los umbrales de alerta o modular la respuesta del sistema de asistencia, ofreciendo intervenciones más precisas y menos intrusivas.

Otro uso potencial es el desarrollo de sistemas de alerta preventiva basados en la detección de eventos sonoros externos relevantes, como sirenas, bocinas o alarmas. Dado que estos sonidos están diseñados para captar la atención del conductor, su detección automática puede actuar como un mecanismo de respaldo en situaciones donde el conductor no los perciba adecuadamente —ya sea por fatiga, interferencia acústica o aislamiento del habitáculo—, generando advertencias visuales que refuercen la percepción del riesgo.

Además, la capacidad de adaptar dinámicamente los modelos a nuevas condiciones acústicas o perfiles de usuario mediante estrategias como la adaptación incremental abre la posibilidad de aplicaciones altamente personalizadas. Un sistema que reconoce mejor las voces de los ocupantes frecuentes, o que ajusta su comportamiento a entornos urbanos o rurales específicos, puede reducir significativamente la tasa de falsas alarmas y aumentar la precisión operativa, contribuyendo a una experiencia más fluida y confiable para el usuario.

Por último, los recursos generados a lo largo de esta investigación —incluyendo la taxonomía de eventos, el conjunto de datos sintéticos y las grabaciones reales— ofrecen un valor inmediato para la comunidad investigadora e industrial. Estos elementos pueden ser utilizados como referencia para entrenar y evaluar nuevos

modelos, establecer estándares de benchmarking entre arquitecturas o servir como base para estudios centrados en el análisis del comportamiento humano en contextos de conducción.

Los modelos eficientes desarrollados y evaluados en esta tesis, como la adaptación de la arquitectura YOLO al dominio acústico, abren nuevas posibilidades para su implementación directa en hardware embebido a bordo de vehículos, sin requerir infraestructura adicional costosa ni depender de servicios en la nube. Esta característica resulta especialmente valiosa en contextos donde la conectividad no está garantizada o donde la latencia debe mantenerse al mínimo. En consecuencia, se habilita la integración de sistemas de inteligencia auditiva en una amplia gama de aplicaciones reales, que incluyen vehículos particulares, taxis autónomos, servicios de transporte público, monitoreo de flotas comerciales o soluciones de movilidad compartida. Así, la tecnología desarrollada tiene el potencial de beneficiar a una base de usuarios mucho más amplia y diversa, ampliando el impacto social y económico del trabajo.

No obstante, los avances alcanzados dejan abiertas múltiples líneas de investigación que se presentan como extensiones naturales del trabajo. Una primera dirección clave consiste en incrementar el realismo acústico de las bases de datos sintéticas utilizadas para entrenamiento. Aunque se ha demostrado su efectividad para desarrollar modelos robustos, persisten diferencias en la complejidad espectrotemporal y la distribución estadística respecto a los entornos reales. Esta brecha podría reducirse mediante el uso de técnicas avanzadas de generación de audio, como modelos de síntesis basados en aprendizaje profundo generativo (por ejemplo, GANs o modelos autoregresivos), que emulen con mayor fidelidad la variabilidad, el solapamiento de fuentes y las condiciones acústicas típicas de una cabina vehicular.

En paralelo, la taxonomía de eventos sonoros propuesta puede evolucionar hacia formas más granulares y contextualmente sensibles. La estructura actual se basa en observaciones empíricas y literatura especializada, pero una expansión hacia taxonomías específicas para diferentes tipos de vehículos (eléctricos, comerciales, de transporte público, etc.) permitiría desarrollar modelos mejor adaptados a cada escenario de uso. Aún más prometedor resulta el desarrollo

de taxonomías dinámicas y multimodales, que integren datos provenientes de sensores visuales, acústicos y del propio vehículo, permitiendo que los modelos ajusten sus categorías en función del tipo de trayecto, el comportamiento del conductor o las condiciones del entorno.

En relación con los modelos de detección, una dirección especialmente prometedora consiste en enriquecer la arquitectura basada en YOLO con conocimiento preentrenado en conjuntos de datos acústicos más generales y diversos. Esta integración podría ampliar la capacidad del modelo para reconocer una gama más extensa de sonidos, facilitando una mejor generalización y robustez frente a escenarios acústicos complejos. Asimismo, se abre la posibilidad de desarrollar variantes del modelo que sean capaces de entrenarse con distintos niveles de etiquetado: desde anotaciones completas (con inicio y fin temporales), hasta datos débilmente etiquetados o incluso no etiquetados, mediante enfoques de aprendizaje semisupervisado o auto-supervisado. Esta flexibilidad permitiría escalar el entrenamiento a grandes volúmenes de datos con un coste reducido en anotaciones manuales.

En el campo de la adaptación incremental, una evolución lógica consiste en extender la estrategia ISA más allá del reconocimiento de emociones hacia otras tareas auditivas de relevancia en el entorno vehicular. En particular, su aplicación a la detección de eventos sonoros distractores ofrece un alto potencial práctico. Al igual que ocurre con la variabilidad entre hablantes, las condiciones acústicas de los vehículos —según marca, diseño, aislamiento, uso o entorno— pueden alterar significativamente la distribución y características de los sonidos relevantes. La capacidad de ajustar modelos preentrenados a estas condiciones específicas, utilizando apenas un pequeño conjunto de muestras etiquetadas, permitiría mejorar la precisión sin comprometer eficiencia ni escalabilidad. Esto es especialmente útil en escenarios como flotas comerciales, donde la heterogeneidad entre vehículos o rutas es elevada, y donde una adaptación ligera pero eficaz puede marcar una diferencia crítica en el rendimiento del sistema.

En síntesis, los resultados de esta tesis no solo aportan soluciones técnicas concretas, sino que delinean una hoja de ruta clara hacia sistemas auditivos vehiculares más inteligentes, adaptativos y accesibles. Las herramientas

desarrolladas —desde bases de datos y modelos eficientes hasta estrategias de adaptación progresiva— ofrecen un punto de partida sólido para futuras investigaciones y aplicaciones industriales. El reto de integrar percepción auditiva en vehículos, más allá de ser una frontera tecnológica, representa una oportunidad real para mejorar la seguridad, el confort y la interacción entre humanos y máquinas en la movilidad del futuro.

En última instancia, esta tesis invita a repensar el papel de la audición como canal sensorial en la conducción asistida y autónoma, reivindicando su valor como fuente rica de información para la prevención y la toma de decisiones en tiempo real. Frente a un futuro de movilidad cada vez más automatizada, sensible y conectada, dotar a los sistemas de la capacidad de "escuchar" no solo amplía su horizonte perceptivo, sino que los acerca a una interacción más humana, capaz de asistir de manera activa y contextual al conductor. La audición permite captar indicios que escapan al campo visual, anticipar riesgos y complementar la percepción ambiental con una dimensión temporal única. Integrar esta capacidad no es solo un avance tecnológico, sino también una apuesta por una conducción más segura, inclusiva y consciente del entorno. Es, en definitiva, un paso hacia vehículos que entienden su entorno no solo a través de lo que "ven", sino también de lo que "oyen" y "comprenden".