POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Exploring Deep Learning Techniques to Improve Voice Disorder Diagnoses

Supervisors

Prof. Tania CERQUITELLI

Prof. Gabriele CIRAVEGNA

Dr. Alkis KOUDOUNAS

Candidate

Qingqing LIU

March 2025

Abstract

Voice disorders are a significant health problem, affecting approximately 7% of the adult population each year, and impacting social and occupational health. However, the majority of clinical diagnostic methods continue to rely on invasive techniques, that may cause patient discomfort and high costs, limiting the accessibility of early screening. This highlights the need for automated, non-invasive diagnostic solutions. With the advancement of artificial intelligence (AI) and deep learning, Transformer-based models have shown great potential in voice disorder diagnosis, using self-attention mechanisms to capture long-term dependencies in voice. However, voice disorder diagnosis technology still faces many challenges due to data limitations, variability of voice signals, and the need to improve diagnostic accuracy. Existing methods have limitations in dealing with small, unbalanced datasets, and environmental noise interference, which restricts the generalization ability of the model. To address these issues, this study proposes a Transformer-based, end-to-end voice disorder detection and classification model. We use data augmentation techniques to alleviate data shortages, and improve the robustness of the model under different conditions. In addition, multimodal fusion is incorporated further. Specifically, we design a time-domain-based audio augmentation pipeline with three different augmentation strengths–30%, 50%, and 70%, to examine its impact on generalization. Furthermore, we focus on two different types of voice recordings — sentence reading (CS) and sustained vowel voicing (SV) — we design and evaluate five multimodal fusion strategies at three levels: early, mid, and late. The results show that in the binary classification task, combining multimodal fusion with data augmentation improves accuracy by up to 7% and macro-F1 by up to 8.9% compared to the single-modality baseline. On unseen data, controlling augmentation intensity further boosts accuracy by up to 6.4%, and macro-F1 by up to 23.3%. In the multi-class task, the optimal fusion strategy yields accuracy improvements of up to 13.7%, and macro-F1 gains of up to 22.1%. These results demonstrate the strong generalization ability of the proposed approach, and its potential for non-invasive voice disorder diagnosis.

Acknowledgments

Looking back, time has flown faster than I thought. From studying Electronics and Communication Engineering at Politecnico di Torino, to now completing my Masters in Data Science and Engineering, this journey has been nothing short of an adventure. I still remember my initial days in the AI field— even something as simple as data preprocessing or running a basic machine learning model, felt like an incredibly abstract task. Projects were mostly beyond my grasp, and to be honest, "time went by so slowly" would be an understatement.

But I wasn't alone. A big thank you to my amazing friends Huanxiao and Kuge — you two always had my back, encouraged me when I doubted myself, and made this journey even more enjoyable. I was lucky to work with Francesco and Giacomo, two brilliant Italian guys, who really helped deepen my understanding of AI and research.

Special thanks to Prof.Ciravegna, who met with me every week from the beginning of my thesis to the end. He patiently answered my endless questions, and gave detailed feedback on my thesis — down to every sentence. I am very grateful for his guidance, as well as the support of Prof. Cerquitelli and Dr.Koudounas, who helped me move in the right direction throughout the process.

Additionally, I would like to thank my parents. Their unwavering support, both financially and mentally, gave me the confidence and freedom to devote myself to my studies. No matter what difficulties I encountered, they always stood by me, believed in me, and most importantly - they never stopped being proud of me.

Finally, I would like to borrow the words of Tagore: 'My last tribute is to those who know I am not perfect but still love me.' A huge thank you to all the amazing people who have been a part of this journey!

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

ASR
Automatic Speech Recognition

CNN
Convolutional Neural Network

RNN
Recurrent Neural Network

MFCC
Mel-Frequency Cepstral Coefficient

HNR
Harmonic-to-Noise Ratio

MLP
Multi-Layer Perceptron

DL
Deep Learning

ML
Machine Learning

LSTM
Long Short-Term Memory

SVM

Support Vector Machine

KNN

K-Nearest Neighbor

HMM

Hidden Markov Model

GMM

Gaussian Mixture Model

Vq-Wav2Vec

Vector Quantized Wav2Vec

BERT

Bidirectional Encoder Representations from Transformers

CTC

Connection Temporal Classification

HuBERT

Hidden-Unit BERT

MLM

Masked Language Modeling

AST

Audio Spectrogram Transformer

ViT

Vision Transformer

SpecAugment

Spectrogram Augmentation

Chapter 1

Introduction

Voice disorders, also called voice diseases or voice pathologies, are conditions that affect a person's speech due to abnormalities in the pitch, loudness, or quality of the sounds produced in the larynx. Voice disorders have a significant impact on people's daily lives. According to surveys, an average of 7% of adults suffer from voice disorders each year. Surprisingly, these voice disorders may lead to communication difficulties, reduced work efficiency (an average of 7.4 working days lost each year), and even career changes, with 4% of patients reporting career changes due to voice problems [1, 2]. Voice disorders can result from vocal overuse, tumors, or neurological conditions, and are commonly categorized as organic, functional, or psychogenic [3, 4]. Since these disorders can cause throat pain and discomfort, and even impair swallowing function, and cause breathing difficulties, they seriously affect overall physical health. Therefore, accurate early diagnosis is crucial for effective treatment.

Although traditional techniques such as laryngoscopy [5, 6] plus voice assessment are widely used in clinical practice, they have obvious limitations. First, these methods are invasive, causing discomfort—especially for patients requiring repeated examinations (e.g., cancer patients). Second, these technologies often rely on expensive equipment, and highly specialized operators, which limits their accessibility in resource-poor settings. Third, traditional evaluation methods rely on the subjective judgment of doctors, and may introduce subjectivity in the evaluation results. In recent years, the application of non-invasive methods has also been very extensive, such as computed tomography (CT) [7], which provides detailed images of the larynx and its surrounding structures, but the cost is expensive. These methods are mainly used for diagnosis when symptoms are obvious, rather than as active preventive screening tools, which limits their early detection of voice disorders.

The development of artificial intelligence technology, especially the application of deep learning in the field of audio and sound processing, has provided new possibilities for overcoming the above challenges [8]. By achieving automated,

non-invasive, and efficient diagnosis, deep learning methods can reduce the cost of diagnosis, reduce the demand for professionals, and make detection more accessible. In addition, these technologies can be integrated into portable devices or mobile applications, for active screening and real-time monitoring, providing a new solution for the early detection and intervention of voice disorders.

Recently, transformer-based models have been shown to be effective tools for automatic detection and diagnosis of voice disorders [9, 10]. Their core advantage is the ability to capture long-term dependencies in time series data, which is crucial for analyzing complex voice patterns. Through the self-attention mechanism, transformers can not only effectively process large-scale datasets, but also extract complex patterns that determine voice characteristics. However, due to the complexity and diversity of pathological voice characteristics, this field is still under-researched and some problems remain unsolved. First, voice disorder data is difficult to obtain on a large scale, Transformer is prone to overfitting in low-resource environments, limiting its generalization ability. Second, noise interference in real-world environments (such as hospitals and homes) will weaken its advantage, in training on clean data and affect diagnosis accuracy. In addition, voice data usually combines different vocalization modalities, doctors often ask patients to pronounce vowels '/a/' first, and then evaluate them by reading sentences. In the following paragraphs, we will introduce how we use the advantages of transformers, to solve some challenges in the field of voice disorders.

Data augmentation and generalization In reality, the collection of medical data is often limited by the number of rare cases and the high cost of annotation. This data scarcity problem limits the training and generalization capabilities of deep learning models.

To address this problem, we adopted a comprehensive data augmentation strategy, to artificially expand the scale and diversity of the dataset. By applying transformation methods such as pitch shifting, time stretching, and noise injection, a variety of acoustic changes can be simulated, thereby helping the model learn more generalizable feature representations. The experimental results also prove this point. Using data augmentation strategies alone can achieve a 3% to 4% performance improvement in accuracy and macro-F1 score in voice disorder detection, and a 5% to 6% improvement in the classification.

In addition, the recording conditions of medical voice vary depending on economic costs and equipment availability, and usually involve two different collection methods: one is collected in a standard environment using professional microphones, and the other is obtained using mobile devices. The latter usually exhibits higher noise levels and other distortions, and its distribution may deviate greatly from the controlled clinical environment, making it difficult for the model to maintain stable performance in different devices and environments. In fact, data augmentation

technology not only expands data, but also plays a vital role in cross-domain generalization. By introducing more severe controlled noise and transformation, the model is more robust to data from mobile devices.

In addition, introducing a small amount of mobile-recorded data during fine-tuning, resulted in notable improvements across both detection and classification tasks, demonstrating the model's improved adaptability to real-world acoustic conditions.

Multimodal learning In practice, doctors perform different patient voice assessments, to evaluate different voice properties, such as requiring the patient to read pre-defined five sentences and emitting sustained vowels. Our study addresses this challenge by designing a framework for multimodal voice disorder detection and classification. Specifically, by processing two types of voice data: sentence reading and sustained vowels. We design a unified model to process them together. Three fusion strategies, early fusion, mid-level fusion, and late fusion, are investigated, which are carried out at the data level, feature level, or decision level to effectively integrate cross-modal information.

We empirically demonstrate that, fusing two modalities is beneficial to improving model performance. In particular, using cross-attention on the detection task, fusion alone can achieve an accuracy of up to 88.5%, and when combined with data augmentation, the accuracy is further improved to up to 89.7%. It illustrates the feasibility of multimodal Transformer-based models in clinical applications, and lays a solid foundation for further advancing automatic speech disorder diagnosis.

Summary This study focuses on deep learning-based voice disorder detection and classification, with experiments focusing on data augmentation, multimodal integration, and cross-domain generalization. We demonstrate that data augmentation can effectively improve the robustness of the model, and enhance adaptability under different recording conditions. In addition, multimodal fusion can capture complementary diagnostic features. Finally, we also evaluate pre-trained model selection, and compare detection-based models and non-detection-based methods in multi-class classification.

The rest of this thesis is organized as follows: Chapter 2 reviews the research work in the field of voice disorder diagnosis, including basic medical knowledge, acoustic feature extraction, common augmentation techniques, and the evolution of deep learning techniques for diagnosing voice disorders. Chapter 3 details our approaches, including data processing, model selection, and fusion strategies. Chapter 4 presents experimental results, Chapter 5 discusses our main findings and limitations, and finally Chapter 6 concludes this thesis and highlights potential directions for future research.

Chapter 2

Related Work

In this chapter, I will provide an overview of the relevant research and methodologies that have been applied in the field of voice disorder detection and diagnosis. First, I will discuss in Section 2.1 the medical background of voice disorders, including their epidemiology, classification, and the clinical diagnostic tools commonly used in practice. Following this, I will explore in Section 2.2 various techniques for voice feature analysis, focusing on feature extraction and the use of data augmentation to enhance model performance. Finally, I will review the state-of-the-art classifiers for voice disorder detection and classification in Section 2.3, covering both traditional machine learning techniques and recent advancements in deep learning, such as CNN, RNN, and transformer-based models. In addition, some mixture of expert models are introduced to further enrich the diversity and effectiveness of the classifiers. This chapter aims to provide a comprehensive overview of the tools and methods that form the foundation for the work presented in subsequent chapters.

2.1 Medical Background

Voice disorders affect a significant portion of the population and have a variety of causes, ranging from neurological conditions to voice abuse. Understanding the medical background of these disorders is essential to developing effective detection and diagnostic methods. This section will first explore the epidemiology of voice disorders, focusing on their prevalence and risk factors (Sec.2.1.1). I will then discuss the various types of voice disorders and their clinical classification (Sec.2.1.2). Finally, I will review commonly used clinical diagnostic tools to provide context for the need for more advanced automated detection and diagnostic methods (Sec.2.1.3).

2.1.1 Epidemiology of Voice Disorders

Epidemiological studies of voice disorders are typically examined from three main aspects: prevalence, potential risk factors, and occupational impact.

Early research, such as the 2005 survey by N. Roy et al.[1], which covered more than 1,300 people found that nearly 30% of people had experienced voice disorders in their lifetime. This study highlighted that factors such as voice use patterns and needs, esophageal reflux, chemical exposure, and frequent colds or sinus infections (excluding tobacco or alcohol) have significant impacts on voice disorders. In recent years, N. Bhattacharyya [2] used the data from the 2012 National Health Interview Survey (NHIS) in the United States to analyze and show that voice problems have a substantial impact on work and life. On average, individuals with voice disorders lose 7.4 working days per year, and approximately 4% of people reported changing careers due to their voice issues. Despite the ten-year gap between the two surveys [1, 2], the percentage of adults they surveyed who experienced voice disorders within a year was still very similar, around 7%.

Voice disorders are well-documented to have effects that extend beyond physical discomfort [3, 4]. These disorders often involve vocal cord damage, which can lead to throat pain and discomfort. Additionally, they can also impair swallowing function and cause breathing difficulties, significantly impacting overall physical health. Additionally, the consequences extend to psychological and social dimensions. Individuals may face communication challenges in daily life, and those who rely on their voice professionally—such as teachers, singers, and call center workers, might be forced to change careers [4]. This can even lead to social isolation and contribute to mental health issues like depression and autism.

Epidemiological studies provide us with extensive data on the prevalence and potential risk factors of voice disorders. However, voice disorders are not a single condition, they have different causes and manifestations. To better address this diversity, it is crucial to conduct deeper research into the classification of voice disorders (Sec.2.1.2). Through accurate classification, we can not only identify the unique characteristics of different types of disorders but also provide a basis for developing personalized intervention and treatment strategies, to more effectively improve patients' speech function and quality of life.

2.1.2 Classification of Voice Disorders

Voice disorders, also known as vocal diseases or voice pathologies, are medical conditions involving abnormal pitch, loudness, or quality of the sounds produced by the larynx, affecting speech.

Voice disorders have been classified into distinct categories in several pathological studies [11, 12, 13], based on their etiologies and clinical presentations. Organic disorders include vocal nodules, vocal polyps, and vocal paralysis. Vocal nodules

usually cause hoarseness, accompanied by voice fatigue or loss of voice, and patients may feel tension in the larynx when speaking. Vocal polyps often manifest as hoarseness, laryngeal discomfort, and intermittent voice disorders. Their presence may lead to decreased sound quality and difficulty in pronunciation. Vocal cord paralysis manifests as hoarseness and difficulty in pronunciation, and in severe cases may affect breathing and swallowing functions. Functional disorders include functional voice disorders and vocal overuse. Patients with functional voice disorders may have abnormal pitch, volume, or quality of voice, but no obvious organic lesions. Common symptoms include voice fatigue and laryngeal discomfort. Vocal overuse is mainly manifested as hoarseness, laryngeal pain, and decreased voice control ability, which is usually related to prolonged or improper use of the voice. Psychological speech disorders are speech problems caused by psychological or emotional factors, where the patient's voice may suddenly become hoarse, dysphonia, or complete loss of voice, often accompanied by emotional stress, anxiety, or traumatic experiences, and the voice abnormality may worsen or improve with emotional fluctuations. Psychogenic speech disorders are speech problems caused by psychological or emotional factors.

The classification of voice disorders offers a foundational framework for clinical diagnosis, but achieving accurate detection and diagnosis typically requires the use of various clinical tools to thoroughly assess vocal cord function. While different types of voice disorders may exhibit similar symptoms, traditional clinical tools remain crucial for uncovering their underlying causes. In the following section 2.1.3, I will present the primary clinical tools and techniques currently employed for detecting and diagnosing voice disorders.

2.1.3 Clinical Diagnostic Tools

Traditionally, the detection and diagnosis of voice disorders relied heavily on invasive procedures. Invasive medical devices, such as direct and indirect laryngoscopes [5, 6], are used to visually inspect the state of the vocal cords. Although these methods provided direct observation of vocal cord movement and morphology, they often caused discomfort, required local anesthesia, and were associated with high costs. Moreover, these procedures required professional operation and specialized equipment, limiting their accessibility.

As technology advanced, non-invasive methods for diagnosing voice disorders became more prominent. Techniques like Computed Tomography(CT) [7] and Magnetic Resonance Imaging (MRI) [14] provided detailed images of the larynx and its surrounding structures, offering a less invasive alternative to traditional methods. These methods, though more comfortable, still involve expensive and complex equipment. High-frequency voice wave imaging, a non-invasive and real-time technique, also emerged but remains mainly in research and clinical trial

stages.

In recent years, acoustic analysis technologies based on machine learning and deep learning, have become increasingly accurate in identifying and classifying various types of voice disorders by collecting and analyzing large amounts of voice data. These advancements have significantly improved diagnostic accuracy. Furthermore, with the rise of mobile devices, the development of telemedicine and mobile applications has enabled patients to record their voices remotely and receive preliminary diagnoses, reducing the need for hospital visits. This trend not only facilitates early detection of voice disorders but also paves the way for personalized treatment plans.

In the following sections, we will explore the specific applications of acoustic feature extraction, machine learning, and deep learning in the diagnosis of voice disorders.

2.2 Voice Feature Preprocessing

This section introduces key techniques for voice feature extraction (Sec.2.2.1) and data augmentation (Sec.2.2.2), which are essential for the accurate detection of voice disorders.

Voice feature extraction focuses on extracting meaningful features from voice signals, such as MFCC and prosodic features, which are essential for distinguishing healthy voices from pathological voices and classifying voices of different pathological types.

Next, we will explore voice data augmentation, a method used to enhance the performance of machine learning systems. These systems, often referred to as models, are computer programs designed to learn from data and make predictions or decisions based on that learning. Voice data augmentation can enhance these models by generating new training examples from existing recordings, thereby expanding the dataset's size and diversity. This may lead to improved reliability and effectiveness of these models.

2.2.1 Voice Feature Extraction

When processing voice signals, digitization is essential for effective analysis. This section is organized into two main parts: first, we will review the key terms and techniques related to traditional feature extraction for voice data (Sec.2.2.1) [15], and then we will delve into models used for feature extraction (Sec.2.2.1).

Traditional feature extraction methods

Traditional voice features are generally categorized into three types: time-domain features, frequency-domain features, and time-frequency domain features. Here, we focus on the acoustic parameters commonly used in clinical acoustic analysis.

Time-domain features include fundamental frequency (F0), jitter, and shimmer. F0 represents the pitch of the voice, while jitter and shimmer reflect the stability and consistency of the speech signal.

Figure 2.1 displays the speech waveform of a healthy individual reading a sentence. The horizontal axis represents time, while the vertical axis indicates amplitude. The variations in volume throughout the utterance are readily observable, with pauses and emphasis in the sentence clearly reflected in the changes in amplitude. The periodic fluctuations of F0 correspond to the speech tone, demonstrating good overall stability and coherence. Additionally, the irregularities in the waveform suggest minor variations in frequency and amplitude, highlighting the presence of jitter and shimmer.

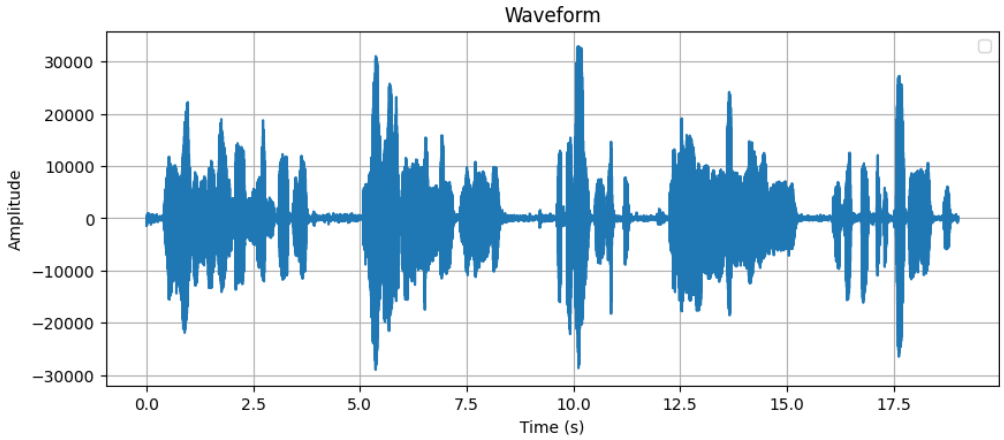


Figure 2.1: Waveform diagram

Note: The diagram shows the time domain characteristics of a audio signal, and illustrates the amplitude variation over time for a healthy individual reading a sentence.

Research has shown that significant differences in these acoustic parameters can be observed between individuals with and without speech disorders [16]. For example, increased jitter and shimmer may indicate conditions such as Parkinson's disease or vocal cord nodules, which affect voice stability. These time-domain features have been shown to be useful in classifying various laryngeal diagnoses, including voice disorders caused by vocal cord polyps, unilateral vocal cord paralysis, etc. [17]. When combined, these acoustic measures can significantly enhance a classifier's ability to differentiate between a healthy larynx and various laryngeal

pathologies. Additionally, by comparing parameters such as jitter and shimmer, it is possible to assess which treatments may offer advantages in improving vocal cord function and voice quality [18].

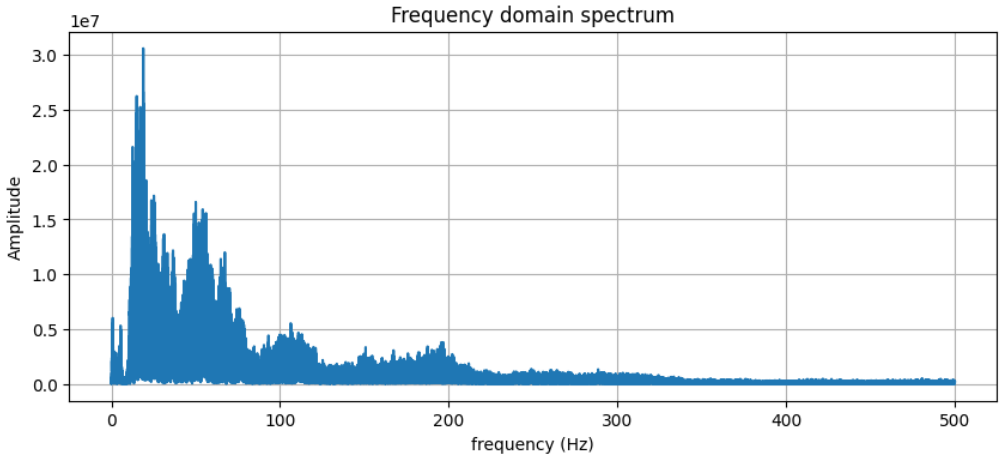


Figure 2.2: Spectrum

Note: The figure illustrates the energy distribution across different frequencies in the audio signal of a healthy individual.

Frequency-domain features analyze signals from the perspective of spectral characteristics. Figure 2.2 shows the speech spectrum of a healthy individual reading a sentence. The figure reveals the energy distribution of different frequency components in the signal, with particularly strong energy observed in the low-frequency range, which is closely related to the position of the fundamental frequency and the formants. This spectrum is instrumental in evaluating the sound quality and timbre characteristics of the voice.

Commonly used clinical features include formant frequency and HNR [15]. Formants are critical for distinguishing vowels and assessing articulatory function, and changes in them often indicate vocal cord dysfunction. In [19], formant analysis has been used to distinguish between healthy and pathological voices, and in [20] has been used to distinguish between pathological types. HNR is frequently used to assess the balance between harmonics and noise in speech signals, providing insight into the overall quality of speech. Studies have shown that HNR is a sensitive indicator of changes in speech quality and is therefore useful for detecting subtle pathologies [21, 22].

Time-frequency domain features, such as the MFCC, wavelet transform and so on, effectively combine temporal and spectral information, making them well-suited for capturing dynamic and non-stationary speech patterns.

MFCCs represent the short-term power spectrum of sound, providing a compact

representation of the spectral characteristics of speech. Figure 2.3 illustrates the MFCC features extracted from a healthy individual's speech. These coefficients capture the spectral characteristics of the speech signal, particularly the formant information, which is crucial for understanding the overall quality of speech. MFCCs exhibit smooth and stable characteristics in healthy speech, making them vital for speech recognition and classification tasks. Their design simulates human auditory perception, allowing for enhanced modeling of voice quality and dynamic speech patterns. For further details on MFCCs, refer to [23]. Wavelet transforms, on

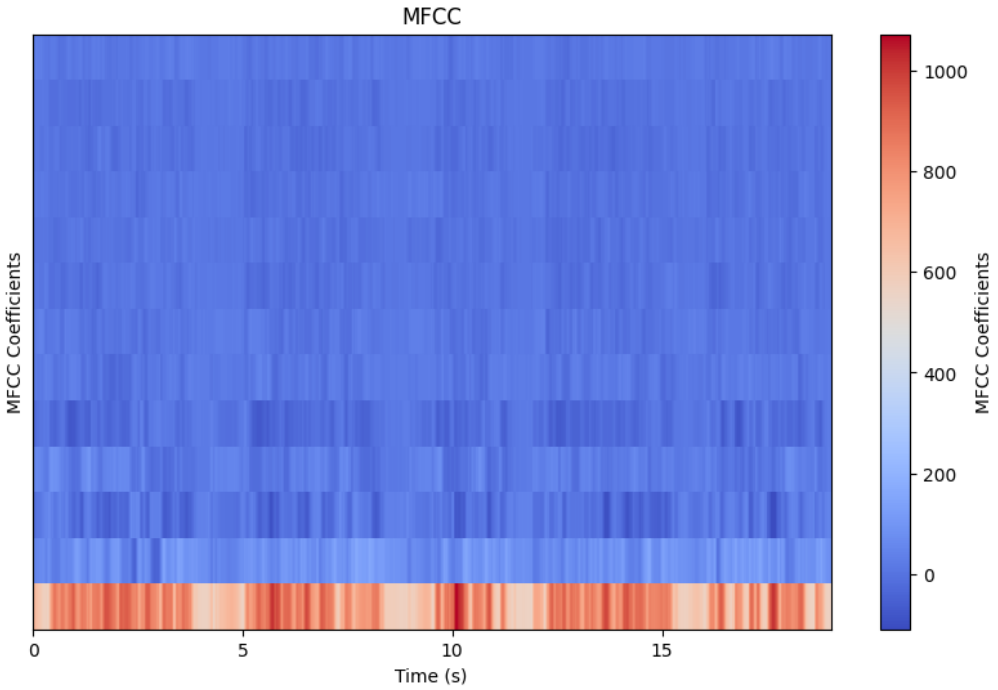


Figure 2.3: MFCC feature map

Note: The figure shows the voice characteristics of healthy individual reading sentences, simulating the frequency characteristics perceived by the human ear.

the other hand, analyze signals at multiple scales and frequencies, allowing for identifying variations in speech over time. For detailed explanation of this technique, see[24]

Among these, MFCC has shown great promise in both voice recognition and pathology detection by simulating the human auditory system. Studies demonstrate that when MFCC is paired with certain classifiers, it significantly enhances the identification of complex voice disorders [25, 26]. One notable study combined time-domain features, like the zero-crossing rate, with frequency-domain features, such as formant frequencies, and time-frequency features like MFCC, achieving

higher accuracy and sensitivity in vocal cord disease classification [27]. Additionally, wavelet transforms have been employed in various models, including MLPs, to distinguish between nodules and Reinke's edema [28]. Further exploration of both continuous and discrete wavelet forms has improved classification accuracy in voice pathology when used in conjunction with MLP [29].

In practice, computer vision models (like CNNs) usually process three-channel (RGB) images [26]. The first three MFCC dimensions are mapped to the red, green, and blue channels respectively, and then these values are normalized to 0-255 pixels, and finally combined into an RGB image. This method retains more time-frequency information than a single grayscale image, while meeting the processing advantages of CNNs for multi-channel data.

While traditional methods for extracting acoustic features have been shown to be effective in identifying various voice pathologies, they often rely heavily on data preprocessing steps and manually computed features, which limits their flexibility and adaptability. In contrast, deep learning [8] has demonstrated its powerful feature extraction capabilities in medical image interpretation, pathology diagnosis, and gastroenterology, reducing the strong need for manual feature engineering.

Model-based feature extraction method

Model-based feature extraction is good at processing high-dimensional complex data, and it can identify subtle and nonlinear patterns that are difficult to capture with manual feature extraction.

CNNs automatically learn hierarchical features from input data, making them highly effective for tasks like image and audio classification. CNN architectures come in three main types: 1D, 2D, and 3D, where "D" refers to the dimensionality of the input data (see Figure 2.4) [30, 31]. The 1D CNN is optimized for one-dimensional data such as time series and audio signals, applying convolution operations along the time dimension to effectively capture local temporal features and patterns, which is particularly useful for tasks like speech and sound classification [32]. The 2D CNN extends this concept to two dimensions, making it ideal for processing images and spectrograms generated from audio signals. When an audio signal is converted into a two-dimensional spectrogram, the 2D CNN can analyze the spatial relationships among different frequency components over time, enhancing its ability to detect key patterns in audio classification tasks [33]. Finally, the 3D CNN is designed for data that has both spatial and temporal dimensions, making it suitable for analyzing sequences of images or volumetric data, thereby capturing complex patterns across time and space.

In summary, CNNs play a crucial role in feature extraction, enabling the identification of a wide variety of audio signals and their respective features [34].

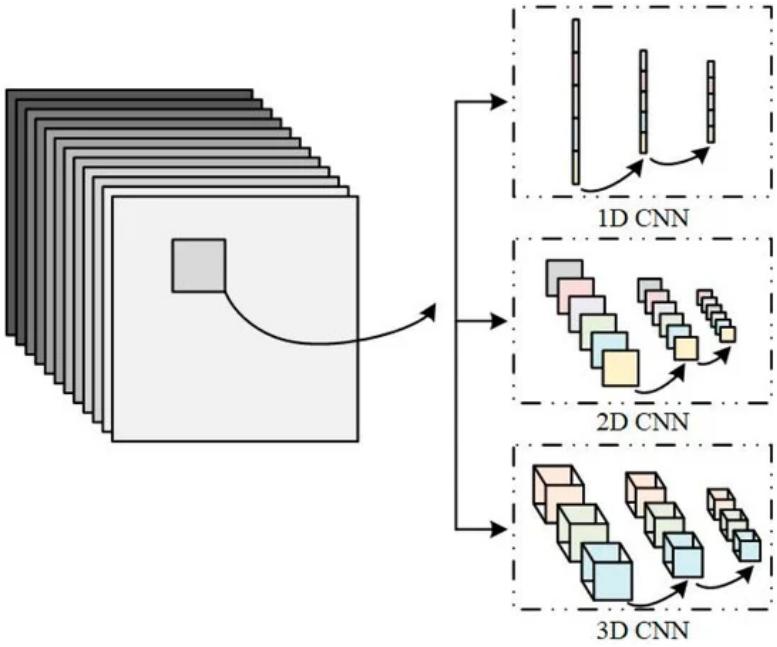


Figure 2.4: 1D, 2D, and 3D CNN [30]

For example, a CNN can convert a one-dimensional speech signal into a two-dimensional spectrogram, extract local time-frequency features through convolution operations, and capture anomalies such as formant shifts and irregular frequencies [35, 36, 37, 38]. These features are extracted layer by layer through multiple layers of convolution, helping the model understand time-frequency information at different scales, the overall architecture of the CNN can be seen in Figure 2.6 in Section 2.4.1.

An RNN or LSTM is good at capturing temporal dependencies in speech, especially dynamic changes such as jitter and pitch fluctuations, which are difficult for manual methods to accurately capture in the long-term temporal context [37, 38].

Transformer captures global and local features at the same time through the self-attention mechanism, analyzes the entire speech signal, and can identify long-term dependencies and complex patterns across time periods, surpassing traditional manual methods [9, 39, 40, 41, 42, 43].

Compared with manual feature extraction, model-based feature extraction is more automated and adaptable, can learn complex features from data, and is particularly suitable for processing dynamic changes in speech signals. This makes deep learning models perform well in sound barrier detection and classification tasks, and can more accurately identify and classify pathological voice signals.

2.2.2 Voice Data Augmentation

ML and DL techniques play an important role in automatic voice pathology detection and classification, but these methods usually require a large amount of training data. However, the high sensitivity of patient medical history makes it challenging to collect large numbers of samples, as this information is often considered important private data. As a result, the available datasets are often limited [8]. To address the challenge of limited datasets, data augmentation techniques have gained widespread attention in recent years and have become a crucial method for improving model performance. Data augmentation refers to a set of techniques that artificially enlarge the size of a training dataset by creating modified versions of existing data samples. This process enhances the diversity of the training data without the need for additional real-world samples, which is particularly valuable in fields such as medical diagnostics, where acquiring new data can be costly or raise ethical concerns. By augmenting the dataset, models can better generalize to unseen data, increasing their robustness and accuracy.

Currently, commonly used audio data augmentation methods include three time domain methods (noise addition, pitch shifting, and time stretching), one time-frequency domain method (Spectrogram Augmentation), and two vocoder-based methods (HNR) modification and glottal pulse length modification) [44, 45, 46].

Time-Domain Methods:

- **Noise addition:** This technique improves the model's robustness in noisy environments by adding background noise to speech samples.
- **Pitch shifting:** Alters the pitch to generate a variety of tone samples, allowing the model to adapt to different speech inputs.
- **Time stretching:** Adjusts the speech duration and speaking speed without changing the pitch, enabling a better representation of diverse speech patterns.

Time-Frequency and Vocoder-Based Methods:

- **SpecAugment:** Randomly covers part of the time or frequency area on the spectrum to enhance the model's learning ability for speech features.
- **HNR modification:** Adjusts the ratio of harmonics to noise, improving the model's adaptability to changes in sound quality.
- **Glottal pulse length modification:** Simulates different pronunciation styles and enriches the diversity of speech samples by changing the duration of the glottal pulse.

These techniques can be used individually or combined to achieve good results [10]. By retaining the labels of pathological and healthy voice samples in existing datasets, they generate new voice samples on the original data. As a result, they significantly improve the model's robustness and accuracy in processing pathological sounds, thereby contributing to more reliable detection and classification of voice pathologies, even when the available data are limited. As data augmentation techniques continue to advance, they hold the potential to further improve the performance of speech pathology detection models, making them more effective and adaptable across various clinical settings.

2.3 Advanced tools for Voice Disorder Detection and Classification

Detecting and diagnosing voice disorders has transitioned from traditional clinical assessments to modern computational methods involving ML and DL classifiers. Initially, traditional approaches relied on gathering detailed medical histories and conducting physical exams, often using invasive techniques like laryngoscopy. While these methods were effective, they required specialized equipment, and skilled professionals, and could cause discomfort to patients. With advances in technology, non-invasive methods based on ML and DL classifiers are becoming increasingly important. These models are trained on large datasets of voice recordings and can automatically learn patterns that distinguish between healthy and disordered voices. This shift offers improved diagnostic accuracy, accessibility, and scalability, addressing the limitations of traditional methods while providing faster and more reliable results for detecting voice disorders.

2.3.1 Machine Learning

Traditional machine learning methods have made significant progress, which provides a solid foundation for the application of deep learning methods. Especially in early-stage research where computing resources are scarce. These classifiers have demonstrated their effectiveness in the detection and classification of various voice disorders, yielding promising results in practical medical environments.

SVM is a commonly used supervised learning algorithm, mainly used for classification and regression tasks [47]. The basic principle of SVM is to find an optimal hyperplane to separate data points of different categories. This hyperplane achieves optimal classification by maximizing the interval (i.e. "margin") between categories. It has proven to be particularly effective for classifying voice disorders, especially when clear decision boundaries exist between normal and pathological voice features [48, 49, 50]. A notable study by R. Behroozmand and F. Almasganj

used SVM combined with genetic algorithms to improve classification accuracy for vocal cord paralysis, underscoring the clinical utility of SVM in diagnosing voice disorders [51]. Despite their success, SVMs require careful hyperparameter tuning and perform best with well-structured data, making them less suitable for large-scale unstructured speech datasets.

KNN is also a simple and effective supervised learning algorithm widely used in classification and regression tasks [52]. The basic idea of the algorithm is to classify or predict by measuring the distance between new samples and known samples in the training set. Specifically, new samples are assigned to the majority category among their K nearest neighbors. This voting mechanism based on neighboring samples enables KNN to flexibly adapt to different data types and distributions when dealing with problems such as pattern recognition and medical diagnosis. It has been applied to various tasks in early diagnosis systems, such as detecting Parkinson's disease by classifying subtle speech abnormalities [53]. L. Chen and C. Wang demonstrated that KNN can perform even better when used with advanced techniques, showing its versatility in clinical contexts [54]. However, KNN can struggle with large, high-dimensional datasets, which limits its scalability and efficiency in more complex voice disorder applications.

HMM is a statistical model widely used to analyze and model sequence data, especially in speech processing, natural language processing, and bioinformatics [55]. HMM is based on the Markov process, where the current state depends only on the previous state, and contains hidden states (unobservable) and observable states (measurable). Through transition probabilities and emission probabilities, HMM describes the transitions between hidden states and the likelihood of observing a certain observable state in a specific state. The model analyzes observable sequences to infer the most likely hidden state sequence and thus performs well in applications such as speech recognition, part-of-speech tagging, and gene sequence analysis. Which are designed for sequential data and have been an important method for voice signal processing for many years. In voice disorder detection and classification, HMM can effectively capture the dynamic changes and nonlinear characteristics in speech by modeling the sound signal. One study indicates that HMM has been applied to analyze features such as MFCCs and pitch dynamics[56]. For example, Arias-Londoño et al. optimized HMMs to enhance accuracy in diagnosing voice disorders, demonstrating the algorithm's effectiveness in medical applications [57]. However, HMMs tend to perform best on smaller datasets with linear relationships, and their limitations become more evident when applied to larger, more complex datasets.

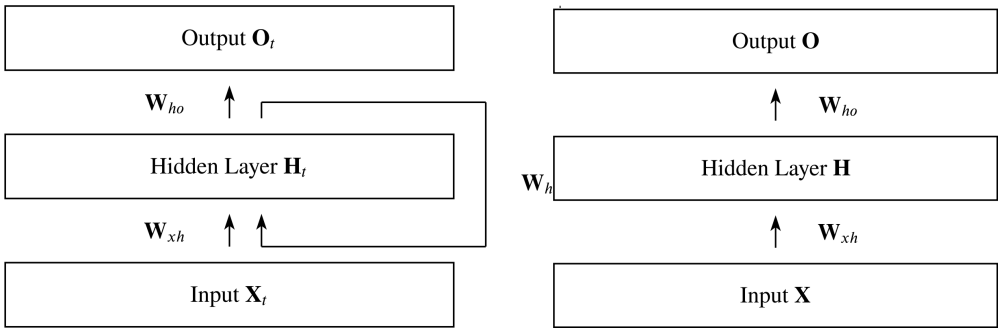
GMM is a mixed probability model that is widely used to describe datasets composed of multiple Gaussian distributions [58]. The model regards the data as a combination of multiple Gaussian distributions, each of which represents a subgroup in the data. By learning the mean, moment, and weight of each

subgroup, GMM can effectively capture the complex distribution characteristics of the data and provide more accurate modeling of the data. In the context of voice disorder detection and classification, GMMs serve as a commonly used classifier that effectively models voice features to distinguish between various voice signals. They have demonstrated considerable accuracy in differentiating between different vocal conditions, including vowel classification and various voice disorders [25, 59]. Despite their ability to model voice feature distributions, GMMs, like GMMs, face challenges when dealing with non-linear patterns in speech data, which are better handled by more advanced techniques like deep learning.

Hybrid models have been developed to combine the strengths of different traditional machine learning methods and overcome the limitations of individual classifiers. One example is the combination of GMMs and SVMs, as demonstrated by Fethi Amara et al., where MFCC features were used, resulting in a 2%–4% improvement in diagnostic sensitivity compared to standalone classifiers [27]. Another study achieved 94.3% classification accuracy by combining wavelet transforms and MFCC features with GMM and SVM classifiers, a significant increase over the 81.4% accuracy achieved by using SVM alone [60].

Although traditional machine learning methods and hybrid models have made significant progress in the field of voice disorder classification, deep learning offers the potential to capture more complex patterns and handle large-scale datasets more effectively. The next section will explore how deep learning models are applied as classifiers for the detection and classification of speech disorders (Sec 2.3).

2.4 Deep Learning for Voice Disorder Detection and Classification



(a) RNN architecture [61] **(b)** Multilayered Perceptron [61]

Figure 2.5: Visualisation of differences between RNN and MLP

Deep learning classifiers have proven to be highly effective in the detection and

classification of voice disorders, which often involve complex, nonlinear, and temporal data. These models, including MLP, RNN and CNN (Sec.2.4.1), Transformers (Sec.2.4.3), process raw speech signals directly, allowing them to automatically learn relevant features, rather than requiring explicit feature extraction methods such as MFCCs or spectrograms.

In this section, we will focus on the commonly used deep learning classifiers for voice pathology detection and classification. We will discuss how these models operate and tackle the task of classifying voice disorders, emphasizing their unique advantages and capabilities in managing the complexity of voice data.

2.4.1 MLP, RNN and CNN in Voice Pathology

MLP

Multilayer PerceptronMLP is a fundamental deep learning architecture composed of multiple layers of neurons, where each neuron in one layer is fully connected to every neuron in the next layer, as illustrated in Figure 2.5 (b).

The forward propagation process in a MLP consists of two main equations [61]. The hidden layer output is computed using the equation :

$$H = \phi_h(XW_{xh} + b_h) \tag{1}$$

where H represents the output of the hidden layer, X is the input vector (features), W_{xh} is the weight matrix connecting the input to the hidden layer, b_h is the bias vector for the hidden layer, and ϕ_h is the activation function applied to introduce non-linearity (such as ReLU or sigmoid). The output layer is calculated by the equation :

$$O = \phi_o(HW_{ho} + b_o) \tag{2}$$

where O denotes the network output, W_{ho} is the weight matrix connecting the hidden layer to the output layer, b_o is the bias vector for the output layer, and ϕ_o is the activation function for the output (for example, softmax).

Although MLPs are generally considered simpler than specialized models like CNNs or RNNs, they have been effectively applied to various speech-processing tasks. This effectiveness arises from the fact that speech features, such as MFCCs, already capture essential time-frequency relationships within the signal. For instance, as demonstrated in Study [62], five different features extracted from the audio signal were input into an MLP to predict emotions, achieving an accuracy of up to 70% with this straightforward architecture.

RNNs and CNNs are two powerful deep-learning architectures widely used in various applications, including the identification and classification of voice pathologies. Their ability to manage complex, high-dimensional data makes them particularly effective for analyzing voice signals.

RNN

Recurrent Neural Networks RNN, on the other hand, are specifically designed for sequential data, such as time series or speech signals [61]. Unlike MLPs (Figure 2.5 (a)), RNNs maintain a memory of previous inputs through recurrent connections (W_h), allowing them to capture temporal dependencies within the data. This makes RNNs particularly suited for tasks where the order of inputs is crucial, such as speech recognition or voice disorder detection.

The hidden state at time step t is computed as follows:

$$H_t = \phi_h(X_t W_{xh} + H_{t-1} W_h + b_h) \tag{3}$$

where H_t is the output of the hidden layer at time t, X_t is the input vector at time t, W_{xh} is the weight matrix from the input to the hidden layer, H_{t-1} is the hidden state from the previous time step, W_h is the weight matrix for the hidden state, b_h is the bias vector for the hidden layer, and ϕ_h is the activation function.

The output at time t is given by:

$$O_t = \phi_o(H_t W_{ho} + b_o) \tag{4}$$

where O_t denotes the network output at time t, W_{ho} is the weight matrix from the hidden layer to the output layer, b_o is the bias vector for the output layer, and ϕ_o is the activation function used for the output.

However, traditional RNNs can struggle with long-range dependencies, which has led to the development of specialized architectures like LSTM networks. LSTMs incorporate mechanisms called gates that regulate the flow of information, enabling the model to learn which inputs to remember and which to forget over longer sequences [38]. This capability is essential for analyzing voice data, where variations in pitch, tone, and rhythm over time can indicate specific pathological conditions [61].

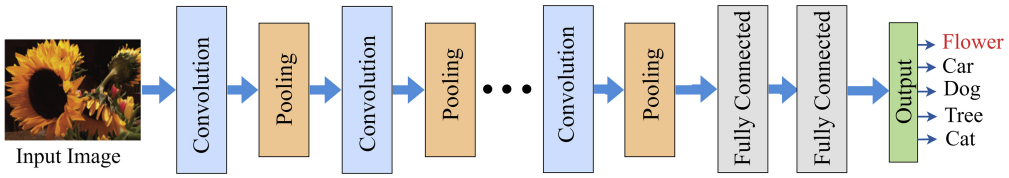


Figure 2.6: Traditional structure of a CNN model.

Note: The diagram shows the typical CNN architecture with varying numbers of convolutional, pooling, and fully connected layers, culminating in an output layer. Source: [63].

CNN

Convolutional Neural Networks CNNs are designed to process data with a grid-like topology, such as images or spectrograms [31]. They use a series of convolutional layers to apply filters to the input data to detect local patterns (See Figure 2.6). The convolution process can be mathematically expressed by the following equation [33]:

$$y = x \cdot h(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) \tag{5}$$

where $h(k_1, k_2)$ represents the filter or kernel applied in the convolutional layer, and $x(k_1, k_2)$ denotes the input matrix. The filter $h(k_1, k_2)$ is first flipped to $h(-k_1, -k_2)$ and then translated by n_1 and n_2, effectively sliding across the input. The negative sign in $h(n_1 - k_1, n_2 - k_2)$ indicates this convolutional operation. Finally, the sum across k_1 and k_2 multiplies the input $x(k_1, k_2)$ by the filter values to compute the output $y(n_1, n_2)$, producing the convolved feature map.

The core idea behind CNNs is to learn a spatial hierarchy of features, where lower layers capture simple patterns (such as edges) and higher layers capture more complex structures (such as shapes or objects). After applying the convolution operation, CNNs often use pooling layers to reduce the dimensionality and increase the model's robustness to input variations. This property makes CNNs particularly effective in extracting features from spectrogram representations of speech signals, enabling them to effectively distinguish between healthy and pathological sounds.

A CNN-based model has been applied to classify pathological voice using both standard voice recordings and electroglottogram (EGG) signals, which capture changes in glottal impedance [35, 36]. These models consist of a feature extraction network and a classification network. Once feature extraction is completed (whether from voice signals or other modalities), CNNs process these features directly, avoiding the need for hand-crafted features, which often lead to information loss. Another CNN-based approach involves the combination of MFCC features with a shallow CNN classifier to detect voice disorders [26]. In this method, after performing basic preprocessing (such as voice activity detection and audio clipping), the MFCC coefficients are passed into a shallow CNN architecture for classification. This architecture is lightweight and computationally efficient, making it particularly well-suited for smaller datasets. However, shallow CNNs may struggle to capture intricate patterns in more challenging datasets. Because these networks primarily identify relatively simple features, they find it difficult to recognize complex relationships and high-order interactions among different features in pathological voice data. As a result, when confronted with diverse and complex pathological voice samples, the classification performance of shallow CNNs may decline.

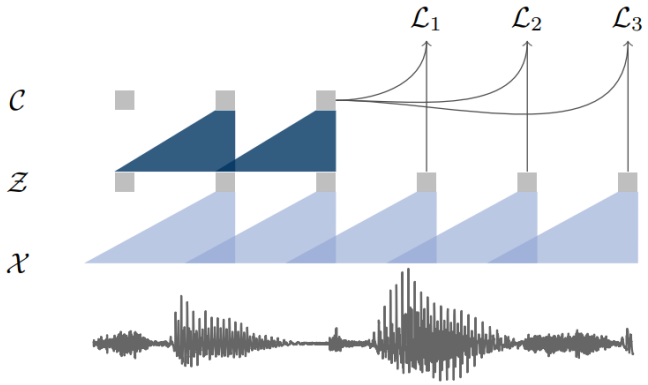
2.4.2 Hybrid models

To address these limitations, Hybrid models that combine CNNs and RNNs offer a more comprehensive solution. While CNNs excel at spatial feature extraction and effectively capture local patterns and structures in data, such as mel-spectrograms, they are limited by their inability to process sequential information. This limitation arises from their reliance on fixed-size inputs and a lack of internal mechanisms to maintain contextual information over time. In contrast, RNNs, particularly LSTM networks, are specifically designed to handle temporal dynamics. They retain information across sequences, allowing them to understand context and relationships that evolve over time. For example, in Parkinson's disease detection, a hybrid CNN-LSTM model was used to classify voice signals after applying a mel-spectrogram as the input feature [37, 38]. First, the CNN component extracts spatial features, and then the LSTM network captures the sequential nature of voice patterns. By leveraging the strengths of both architectures, hybrid models significantly enhance classification performance on diverse and complex voice data. This model achieved higher classification accuracy than traditional machine learning classifiers, such as SVMs and XGBoost [64], on the PC-GITA dataset [65]. While these hybrid CNN-LSTM models have demonstrated enhanced classification accuracy, they come with increased computational costs. Both CNN and LSTM networks are resource-intensive, requiring more time and higher computational power compared to simpler models.

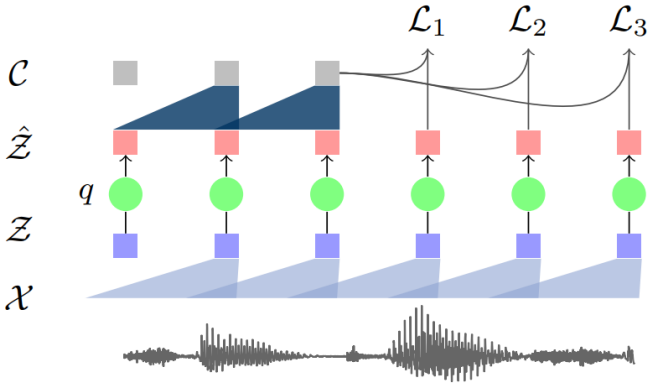
2.4.3 Transformer-Based Methods

The **Tansformers** model, introduced by Vaswani et al. in 2017 [9], utilizes a self-attention mechanism to process input sequences in parallel, allowing it to effectively manage long-range dependencies and complex relationships in data. Unlike MLPs, RNNs, and CNNs, which typically specialize in extracting local and temporal features, Transformers capture both local and global dependencies in sequential data. Through multi-head self-attention, the model can focus on various parts of the input sequence simultaneously, making it especially suitable for handling intricate, multi-level temporal patterns in speech signals. Thanks to these advantages, Transformers have gained widespread adoption in tasks like speech recognition and synthesis, and more recently, in the detection and classification of voice disorders. Their ability to process high-dimensional and complex voice data makes them a valuable tool in medical AI, significantly enhancing the accuracy of pathological voice analysis and diagnosis.

The original **Wav2Vec** model [66], introduced by Baevski et al. in 2019, revolutionized speech representation learning through its specialized CNN architecture, designed for self-supervised pre-training on large-scale unlabeled audio data (See Figure 2.7a). The model primarily consists of a feature encoder that transforms



(a) Wav2Vec [66]



(b) Vq-Wav2Vec [67]

Figure 2.7: Visualisation of differences between Wav2Vec and Vq-Wav2Vec

raw audio input X into latent representations Z, employing convolutional layers to effectively capture local dependencies within the audio signal. Additionally, it incorporates a quantization layer that maps these continuous latent representations into discrete codes C, aiding in information compression while retaining critical audio features. Wav2Vec leverages contrastive learning as its training objective on unlabeled data, optimizing a loss function L that distinguishes between real encoded samples (positive examples) and randomly chosen negative samples. Despite its strong performance on ASR benchmarks [68], the model struggles with capturing long-range dependencies in speech data, primarily because of its reliance on separate steps for feature extraction and classification. This limitation has been gradually addressed by integrating transformer architectures, which unify feature extraction and classification while improving the model's ability to learn more complex, long-range temporal relationships in speech data.

On the other hand, the **Vq-Wav2Vec** model is a variant of the Wav2Vec model [67] (See Figure 2.7b), introducing vector quantization (VQ) to achieve discrete speech representation by using techniques such as Gumbel Softmax [69]. The model works as follows: first, the encoder maps the raw audio X into dense latent representations Z. These representations are then quantized into discrete codes $\hat{Z}$ using vector quantization. The quantized representations $\hat{Z}$ are subsequently aggregated into contextualized representations C, which capture temporal relationships in the audio signal. The training objective for Vq-Wav2Vec involves predicting future time steps, ensuring the model learns temporal dependencies across the audio data. Using these discrete feature representations, models like BERT [39] can be trained on audio tasks, enabling Vq-Wav2Vec to achieve impressive results in tasks that require detailed interpretation of audio signals. This model opens new possibilities for applications like voice pathology detection and classification by providing more efficient and powerful speech representations.

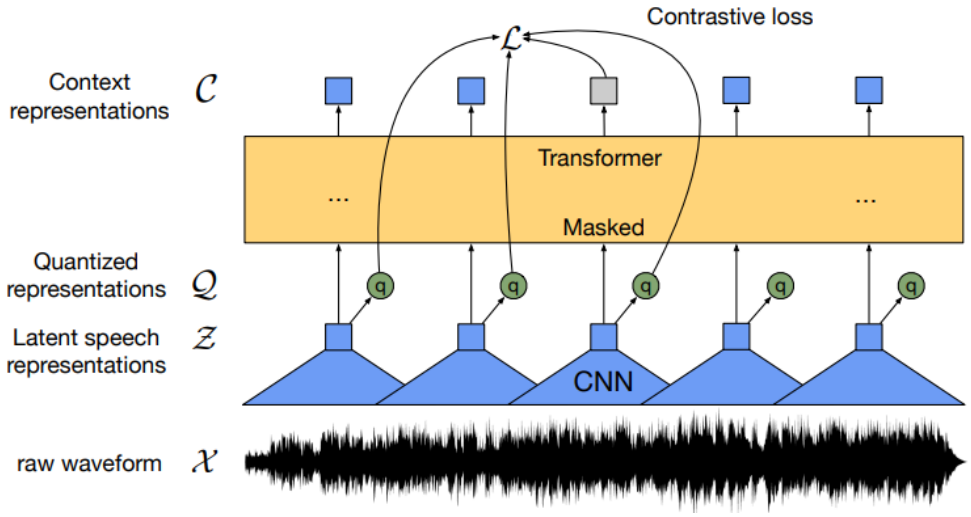


Figure 2.8: Wav2Vec2.0 [40]

Building on these foundations, **Wav2Vec 2.0** retains some key features from the original Wav2Vec while introducing enhancements for improved performance [40] (See Figure 2.8). Specifically, it maintains the self-supervised learning paradigm, where the model learns to extract useful representations from raw audio data X without extensive labeled datasets. However, it enhances this approach by incorporating a gumbel softmax quantization module from Vq-Wav2Vec, which allows for better representation learning through discrete encoding. Gumbel softmax quantization module in Vq-Wav2Vec with a BERT-like Transformer model. Unlike its predecessor, Wav2Vec 2.0 is designed as an end-to-end ASR model that combines feature extraction and classification tasks in a seamless process. The raw speech

signal is initially encoded through a 1D CNN, which is specifically tailored to capture the temporal features of the audio signal. The CNN extracts high-level features from the raw waveform, such as pitch, volume, and timbre, enabling the model to focus on relevant acoustic properties. Some of the latent variables Z are mapped to discrete representations Q through a gumbel softmax quantization module, while others are masked at random locations and fed into a Transformer network to obtain contextual feature representations C. The model computes the self-supervised loss L by contrastive learning over the masked locations. Wav2Vec 2.0 differs from vq-Wav2Vec and earlier Wav2Vec models in that it no longer relies on multiple independent steps (such as feature extraction followed by classification), but instead adopts a unified process of pre-training and fine-tuning. During the fine-tuning phase, a randomly initialized linear layer is usually added on top of the pre-trained Wav2Vec 2.0 model. This layer acts as a classifier to map the high-level features extracted by the model from the audio input to specific target outputs, such as phonemes or words. On this basis, the model is trained with the CTC loss [70] to perform ASR tasks. This end-to-end approach enhances the model's ability to extract meaningful features from speech signals, especially in the medical field where discerning subtle acoustic features is critical for diagnosing voice pathologies.

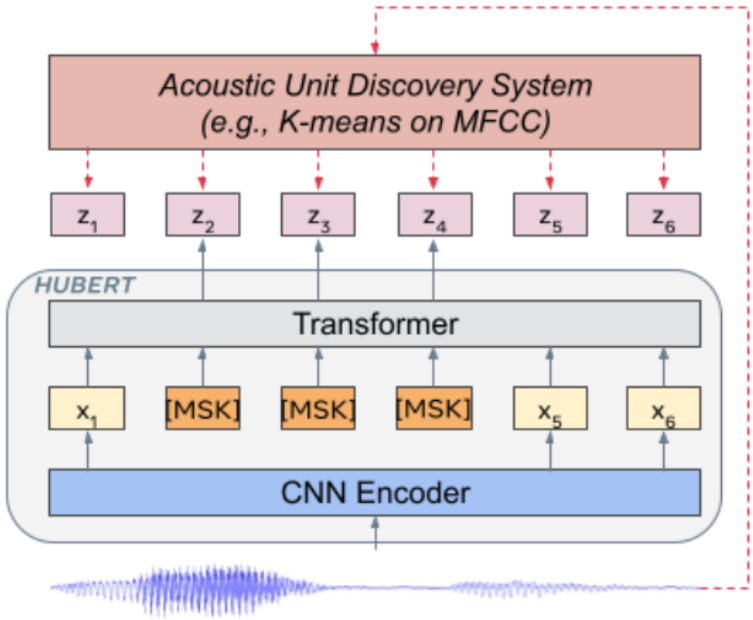


Figure 2.9: Hubert [42]

HuBERT [42] continues the evolution of self-supervised learning initiated by models like Wave2Vec 2.0, but introduces a different mechanism through masked prediction for hidden units. While both models rely on self-supervised learning, the

key difference lies in how the representations are learned (Compare figures 2.8 and 2.9). Wave2Vec 2.0 learns to distinguish between true and negative samples at the feature level using contrastive learning, whereas HuBERT learns through a masked prediction framework, where part of the input is hidden and the model is trained to predict the missing parts. This allows HuBERT to capture more contextual relationships within the speech signal [42]. In HuBERT, let X denote the speech utterance, represented as a sequence of frames $X = [x_1, x_2, \ldots, x_T]$, where T is the total number of frames and each x_t represents a feature vector of the t-th frame. The speech signal is discretized into hidden units $Z = [z_1, z_2, \ldots, z_T]$, where each $z_t \in [C]$ is a categorical variable corresponding to one of C clusters. These hidden units are generated by applying an unsupervised clustering method, such as k-means, to the feature frames. HuBERT's pre-training involves a masked language modeling (MLM) approach, where part of the input sequence is masked, and the model learns to predict the hidden speech tokens z_t from the surrounding unmasked context. This is achieved by masking portions of the input speech X and then using the unmasked frames to predict the hidden units Z. By doing so, HuBERT captures both local acoustic features and long-range dependencies in the speech signal. This framework differs from Wave2Vec 2.0, which focuses on learning continuous latent representations without explicit discretization. In contrast, HuBERT's use of discrete hidden units enhances its ability to model fine-grained acoustic details and capture subtle variations in speech, making it particularly effective for tasks such as detecting pathological speech patterns. The use of the MLM strategy in HuBERT allows the model to infer missing information from the broader speech context, promoting a deeper and more nuanced understanding of the signal.

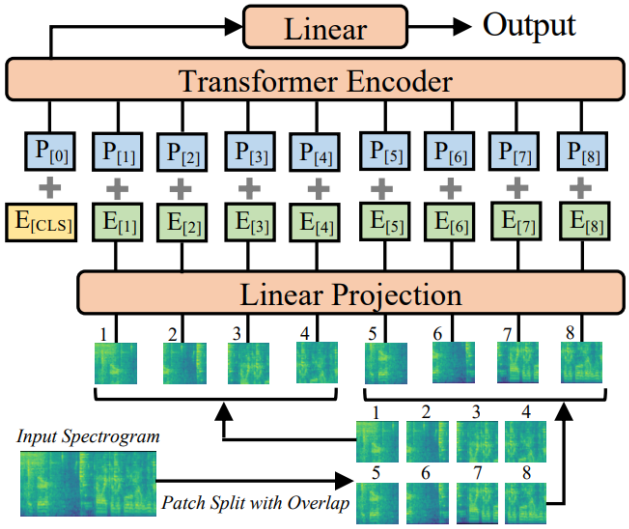


Figure 2.10: AST [41]

Further expanding on transformer-based methods is the **Audio Spectrogram Transformer AST)** [41]. As illustrated in **Figure 2.10**, the AST model introduces attention-based layers from the Vision Transformer (ViT), and replaces traditional convolutional layers. The input to AST is a log-Mel spectrogram, which is split into multiple overlapping patches. These patches are linearly projected into embeddings (denoted as $[P_1] - [P_8]$) and are then passed into a transformer encoder. A special classification token $[E_{\text{CLS}}]$ is appended to the sequence of embeddings, and position encodings $[E_1, E_2, \ldots, E_8]$ are added to the patch embeddings to provide location information. The resulting sequence $[E_{\text{CLS}}, P_1, P_2, \ldots, P_8]$ is fed into the transformer encoder, which utilizes multi-head self-attention to capture both local and global contexts across all layers. By capturing long-range dependencies effectively, AST excels in audio classification tasks that require modeling of complex global patterns.

While AST eliminates CNNs, it reintroduces log-Mel spectrograms as inputs, which are processed entirely through the self-attention mechanism, allowing it to capture both local and global contexts. AST's ability to handle long-range dependencies across all layers makes it particularly effective for audio classification tasks. Additionally, by pre-training the ViT on the ImageNet dataset [71], AST benefits from knowledge transfer from the image to audio domain. This significantly improves performance compared to PSLA model[72], especially in tasks like sound event detection and classification, where capturing detailed global information is crucial..

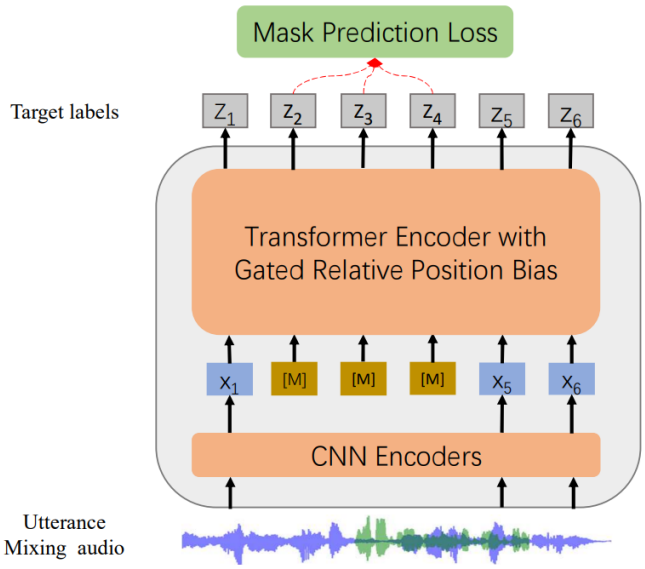


Figure 2.11: WavLM [43]

Moreover, **WavLM** [43], a large-scale self-supervised pre-trained model, takes

these developments even further by providing a comprehensive framework for full-stack speech processing tasks, as shown in **Figure 2.11**. WavLM first processes the input audio signal using CNN encoders to extract feature representations (denoted as $[X_1] - [X_6]$). These features are then passed into a transformer encoder that incorporates gated relative position bias to better handle sequential information in the speech signal. During pre-training, WavLM employs noise masking and time-segment reconstruction tasks. Mask tokens $[M]$ are inserted into the input sequence, and the model is trained to predict the masked portions (e.g., $[Z_2] - [Z_4]$) based on the surrounding context. This pre-training strategy, combined with a mask prediction loss function, enhances the model's robustness in handling noisy or mixed speech inputs.

The flexibility of WavLM across different tasks shows great potential in handling diverse speech-related medical applications, including the detection and classification of voice disorders.

End-to-end architectures like Wav2Vec 2.0 highlights the critical advantage of these models: by integrating feature extraction and classification within a single process, they minimize the need for multiple independent steps and deliver superior performance. In medical diagnostic tasks, where precise acoustic feature extraction is crucial, these models provide a robust foundation for detecting subtle patterns in pathological voice, opening new possibilities for the early detection and monitoring of voice disorders.

2.4.4 Multimodal Approaches in Medical AI

In medical AI, particularly for tasks like detecting Alzheimer's disease, multimodal machine learning—which integrates different data types like speech, text, and images—has shown great promise. Transformer-based models have been especially effective in combining these diverse inputs for more accurate diagnoses [73, 74]. Multimodal data fusion can be achieved through three strategies (See Figure 2.12):

Early fusion:

In early fusion, input data from different modalities are concatenated, and the resulting vector is treated as a single input.

This means the model does not distinguish between the modalities from which the features originated. For example, Thung et al. used a joint fusion approach for combining PET and MRI images [75], and their approach can be classified as early fusion because they concatenated clinical and imaging features into a single feature vector before inputting it into the neural network. [76].

As emphasized in the study by S.R.Stahlschmidt et al.[74], early fusion is relatively easy to implement because it does not require separate modeling of

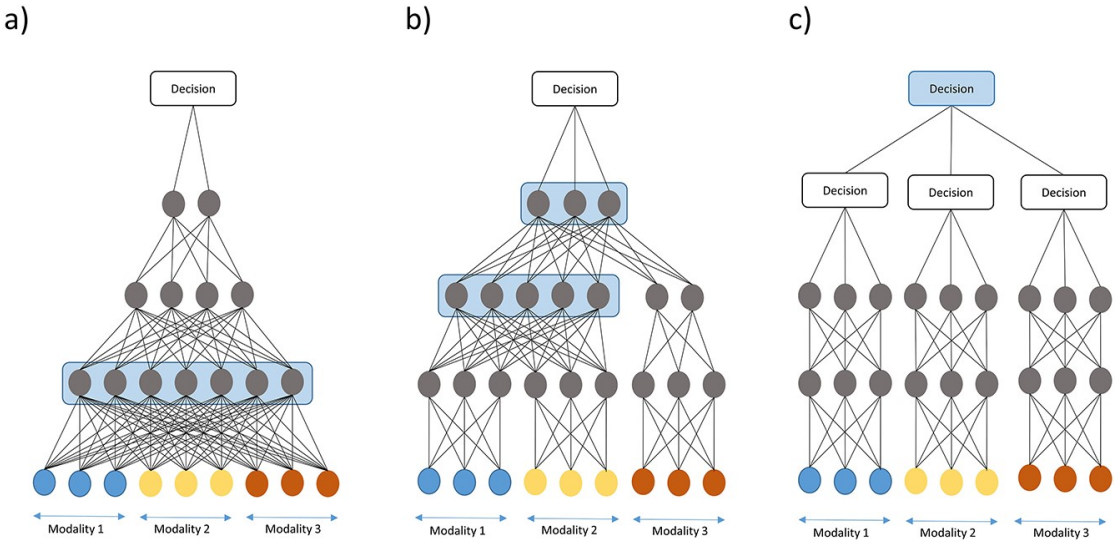


Figure 2.12: (a) Early fusion strategies (b) Mid-term fusion strategies (c) Late fusion strategies [74]

individual modalities. However, it has a weaker ability to capture deeper correlations between modalities and is more suitable for situations where the modalities are quite homogeneous, such as combining different types of imaging data.

Late fusion:

Instead of combining the original data or features, late fusion trains separate models for each modality and uses an aggregation function to combine their predictions to make a final decision.

Common aggregation functions used in the late fusion [74]: **Average**, which is a majority vote on the average of the predicted probabilities of each model, where each model makes a classification prediction and the final classification is the class with the most votes; **Weighted voting**, where each model's prediction is weighted based on factors such as accuracy or confidence, and the class with the highest weighted score is selected; **Meta-classifier**, where a trained model uses the predictions of each individual model as input and learns how to best combine these predictions to provide a final decision. After the aggregation step, classification is determined by evaluating the combined or weighted predictions to determine the final class or output.

In one study [10], A. Koudounas et al. added a late fusion approach into a framework, where separate models were trained on voice vowel audio data and sentence reading audio data. The final decision was made by selecting the output from the model with the highest confidence, which was estimated by the entropy

of the predicted probabilities. This approach effectively leveraged the unique characteristics of each data type, leading to better performance compared to single-model methods.

While late fusion is suited for handling heterogeneous modalities [74], it cannot capture interactions between features from different modalities, making it more appropriate for different modalities with low correlation.

Mid-term fusion:

This approach first extracts features from the input of each modality, converts the raw data into a feature vector, and then fuses the features of different modalities together and merges them into the final model as input. [74].

A practical example of this approach is detecting dementia using both speech and text data. In recent studies, log-Mel spectrograms and MFCCs (features from audio) are processed through a ViT, while text data is analyzed by BERT [77]. The model uses a cross-modal attention mechanism to dynamically focus on the most important features from each modality, leading to better results. Additionally, a gated multimodal unit [78] ensures the model emphasizes relevant information and ignores irrelevant data. This is crucial in medical diagnoses, where different data types (e.g., speech patterns or text) may provide varying levels of importance at different stages of the disease.

The benefit of intermediate fusion is that it allows the model to capture the unique characteristics of each modality and explore potential correlations between them at the feature level. However, because the feature dimensions of different modalities can be highly unbalanced, careful processing is required to prevent information loss [74].

2.5 Conclusion

In summary, existing research on voice pathology detection and classification covers a wide range of methods from traditional machine learning to modern deep learning.

Traditional methods such as SVM and KNN rely on manual feature extraction. Although they have achieved some success in the early stage, they have difficulty in handling complex nonlinear voice features. With the rise of deep learning, methods such as CNN, RNN, and Transformer have shown better performance in voice pathology identification through automated feature extraction and powerful pattern recognition capabilities. However, deep learning still faces some challenges, such as dependence on large-scale data and unstable performance in small sample scenarios.

In order to cope with the problem of data scarcity, data enhancement techniques such as noise addition and time stretching are widely used to improve the robustness of the model. In addition, multimodal fusion methods are expected to further

improve the accuracy of diagnosis by combining different types of voice data, such as monosyllabic pronunciation and sentence reading combined in this study.

Based on these observations, this study proposes an end-to-end model based on deep learning, combined with multimodal voice data, to improve the accuracy and robustness of speech pathology detection. The following sections will introduce the design and experiments of the model in detail.

Chapter 3

Methodology

This chapter introduces some comprehensive approaches we mainly use, to solve the problem of data scarcity, to explore the application of multimodal data in the detection and classification of voice pathologies.

Section. 3.1 describes the data acquisition and processing process in detail, including the characteristics of IPV and New datasets, collection methods, and multiple preprocessing techniques to ensure data consistency, reduce noise, and address data imbalance. Specifically, we regularize the data by truncating the audio length and normalizing the input, while mapping category labels in multi-classification tasks to unify data form, and ensure fair and reliable experimental evaluation. In addition, to address the challenges of data sparsity, multiple data augmentation techniques are designed and implemented to improve the model's adaptability to diverse scenarios.

Section. 3.2 examines the selection of three different architectures: MLP, 2D-CNN, and Wav2Vec2.0. Wav2Vec2.0 as an end-to-end framework, outperforms traditional MLP and 2D-CNN by extracting hierarchical features directly from raw audio, making it well-suited for tasks with limited labeled data. The comparison provides a theoretical and practical foundation for model design in subsequent experiments.

Section. 3.3 introduces early, mid-level, and late fusion strategies, to integrate complementary information from different modalities. These strategies address the limitations of single-modal analysis, and they are systematically evaluated using mathematical formulations, architecture diagrams, and experimental results, highlighting their effectiveness in voice pathology tasks.

Through strict preprocessing, reasonable model selection, and innovative multi-modal fusion strategies, this chapter constructs a complete method framework to support subsequent experiments and analysis.

	Healthy	Pathological	CS	SV	$T(s)$
IPV	362	672	517	517	12.95
New	58	158	108	108	19.91

Table 3.1: Summary of dataset characteristics. *Note: Healthy and Pathological represent the number of healthy and diseased samples, respectively. CS indicates sentence reading samples, SV represents syllable articulation samples, and $T(s)$ is the average audio duration in seconds.*

3.1 Data Acquisition and Processing

3.1.1 Datasets

IPV The Italian Pathological Voice (IPV) dataset is a novel and diverse resource, designed specifically for voice pathology research, currently unpublished and introduced in [10]. Collected from participants in Italian otolaryngology and voice therapy clinics, the dataset includes both healthy individuals and patients with different pathological types of voice disorders. All recordings were conducted under strict standardization protocols in quiet environments, ensuring high-quality samples with a signal-to-noise ratio exceeding 30 dB and a fixed microphone distance of 30 cm. The dataset includes two modalities: sustained phonation of the vowel /a/ (SV) and reading of five phonetically balanced sentences (CS) adapted from the Italian version of CAPE-V [79]. Each sample includes detailed diagnoses of health conditions from experienced physicians. Table 3.1 summarizes the dataset characteristics, with the distribution of healthy and pathological samples, average audio duration, and modal information. Notably, the dataset is relatively small, with only 1034 samples in total, posing challenges for developing robust and generalizable models. The label distribution of pathological samples in the IPV dataset is depicted in Figure 3.1, where certain macro-categories, such as label 3 (Benign Neoplasms), are more frequent, while others, like label 2 (Hemorrhagic Laryngitis), are underrepresented. This imbalance is critical for understanding the challenges in training robust models for multi-class classification tasks.

New dataset It was collected using the mobile device in a less controlled environment, including background noise in the clinic and slight variations in the recording equipment. The samples also come from otolaryngology and voice therapy clinics in Italy, and are annotated in the same way as the IPV dataset, covering both sustained phonation (SV) and sentence reading (CS) modalities, with health status labels. As shown in Table 3.1, the New dataset has fewer samples compared to the IPV dataset, with 58 healthy and 158 pathological recordings. The label distribution for pathological samples, depicted in Figure 3.1, reveals significant

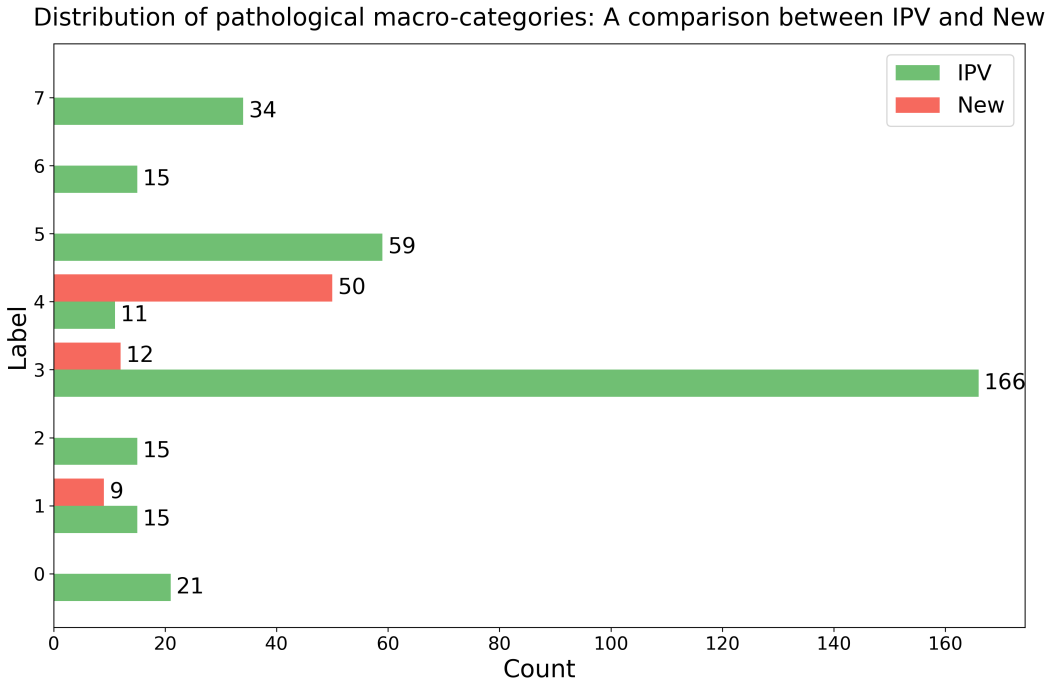


Figure 3.1: Label distribution of pathological samples in a single modality. *Note: The 'label' corresponds to the 'macro-category' field in the dataset, representing high-level diagnostic groupings. Green indicates the IPV dataset, while red indicates the New dataset. For the specific names of labels, please refer to Table 3.2.*

Label	Italian	English
0	disfunzionali presbifonie	Dysfunctional Presbyphonia
1	edema di reinke	Reinke's Edema
2	laringiti emorragie	Hemorrhagic Laryngitis
3	neoformazioni benigne	Benign Neoplasms
4	neoformazioni potenzialmente maligne	Potentially Malignant Neoplasms
5	paralisi	Paralysis
6	spasmodica tremore	Spasmodic Tremor
7	sulcus vergeture	Vocal Cord Sulcus

Table 3.2: Italian-English pathological label mapping.

class imbalance, with some macro-categories being heavily underrepresented. For example, there are only three types of diseases (Label:1,3,4), and compared with IPV, there are 5 types missing. It can be said, that incorporating different recording conditions and different data distributions together, provides valuable insights into how the model performs in real-world scenarios.

The consistent structure and annotation methods between the IPV dataset and the New dataset facilitate integration. The high-quality recordings of the IPV dataset, combined with the environmental diversity of the New dataset, provide a solid foundation for developing and validating voice pathology detection and classification models under controlled and real-world conditions.

3.1.2 Data Preprocessing Techniques

Effective data preprocessing ensures consistency, reliability, and robustness of model training, especially when dealing with multimodal data and small imbalanced datasets. In the following, several preprocessing steps were adopted, including data cleaning, truncation, normalization, label harmonization, and data augmentation.

Data cleaning and Standardization In order to implement the fusion strategy and enable all experiments to be compared on the same data, this study eliminated samples that only contained single-modal recordings (for example, individuals who only has sentence reading records and vowel pronunciation records). Since the number of such samples was extremely small, with only two individuals, their removal had no significant impact on the overall experimental results.

Furthermore, to ensure the consistency of audio duration, and facilitate meaningful comparison, the audio samples in the dataset were truncated to a fixed length: the audio samples of the CS and SV modalities were truncated to 19 seconds, and 18 seconds, respectively. If the duration is shorter than them, it is extended to a fixed length using zero padding. These truncation lengths are designed to cover approximately 90% of the samples in each modality, which can not only preserve the integrity of the audio information, but also effectively reduce the impact of abnormal samples that are too long, thereby improving the generalization ability of the model.

The audio data was then standardized using a predefined processor of the Wav2Vec2.0 framework. The processor first resamples the audio to 16 kHz to ensure compatibility with the framework, while reducing the overhead of computing resources. The feature representation generated after standardization, can effectively capture the key information of the voice signal, providing efficient and consistent input for subsequent model training.

Label Harmonization In the multi-classification task, the consistency of category definitions between the new dataset and the IPV dataset is very important, so a labeling rule based on the macro-category mapping of the IPV dataset was used. Specifically, for the categories in the new dataset that were not defined in the IPV dataset (a total of 16 samples), we removed them to avoid the impact of category inconsistency on model training and evaluation. Through this mapping process,

the uniformity of category definitions between different datasets is guaranteed, enhancing the fairness and reliability of the comparative experiment.

Dataset Partitioning In addition, in order to solve the problem of fewer healthy samples in pathological voice data, we adopted a stratified sampling method in the data segmentation process. This method can separate the representative samples proportionally. The entire dataset is divided into training set, validation set, and test set in a ratio of 8:1:1, to ensure the fairness and repeatability of the experiment. First, the test set is divided using a fixed random seed, and then the training set and validation set are further divided using three different random seeds, to generate multiple data partitions. Finally, by calculating the average performance across these partitions, the robustness and reliability of the evaluation results are improved.

Algorithm 1 Data Augmentation and Feature Extraction

Require: AudioPaths, Labels (L), FeatureExtractor, Max duration (D), Sampling rate (sr), flag, ratio
Ensure: ProcessedFeatures, ProcessedLabels

Define `DataProcessor` with attributes: {FeatureExtractor, D, sr}
function ProcessFile(AudioPath, Label)
    Load audio from `AudioPath` with sampling rate sr
    if flag **is True and** random probability < ratio **then**
        Apply augmentation: {Noise, Time Stretch, Pitch Shift, Combined}
    end if
    Truncate then **Pad** to D
    Extract features using `FeatureExtractor`
    return Processed features, Label
end function

Data Augmentation For the data sparsity issue caused by the small dataset, this study introduced a data augmentation pipeline during the training process.

We perform data augmentation on the training data, and randomly apply four augmentation ratios to **30%, 50%, 70%** of the samples for exploration: adding Gaussian noise, time stretching, pitch shifting, and combined augmentation. The first three belong to voice data augmentation in the time domain (the methods mentioned in Section. 2.2.2), and combined augmentation refers to mixing these three.

Figure 3.2 provides an intuitive visualization of the effects of each augmentation

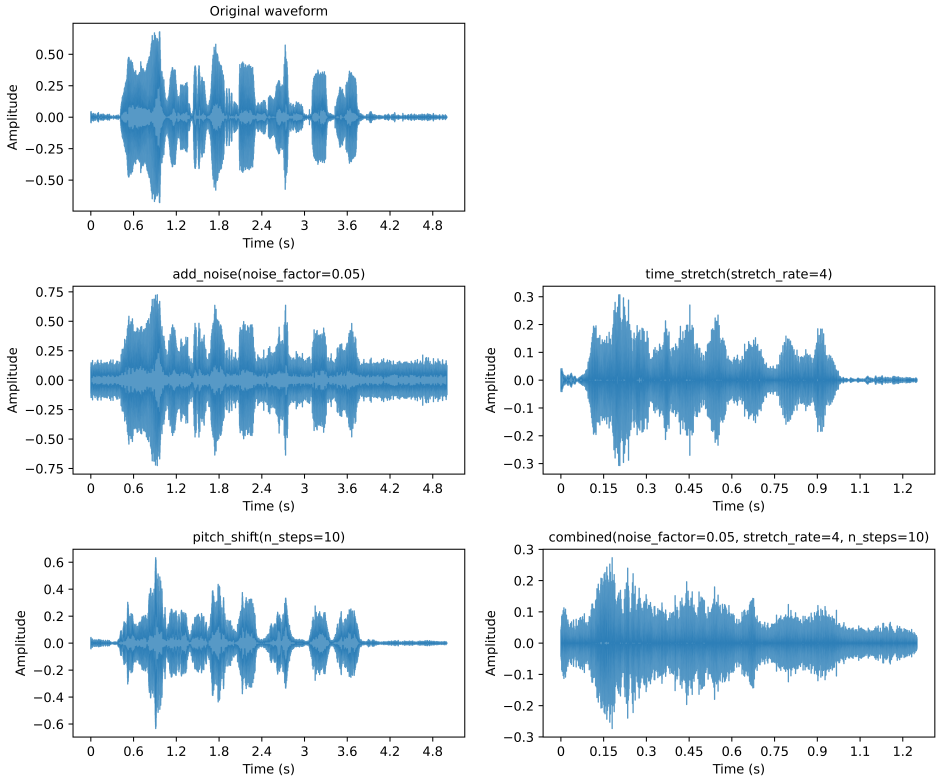


Figure 3.2: Waveforms of the original audio and augmented signals using different techniques. *Note: For visualization purposes, the parameters of augmentation methods (e.g., noise intensity, stretch rate, and pitch shift) were intentionally exaggerated to highlight the effects. In practice, these parameters should be carefully controlled to avoid distorting the original signal.*

method on a waveform, showing the original signal and the augmented versions. It is important to note that while exaggerated parameter values were used for illustration purposes in the figure, the actual augmentation applied during training involved significantly smaller adjustments. This careful control of the parameters ensured that the augmented samples remained representative of the original audio, preserving the essential features necessary for robust model training.

The following algorithm 1 illustrates the augmentation process in pseudocode form. For each audio sample, we first determine whether augmentation is applied.

If we decide to augment, randomly select an augmentation method, and pass the processed audio to the feature extractor.

The remaining unaugmented samples retain their original features to ensure the reliability of the dataset. By balancing the augmented and original data, this method can create a more powerful training set, that can be generalized to various audio scenarios.

Feature Extraction To ensure a consistent input length, all audio samples are first truncated to 18 or 19 seconds according to their modality. If shorter, zero-padding is applied to extend them to this fixed duration. For baseline models such as MLP and 2D-CNN, MFCCs are extracted from the processed audio. In the case of the 2D-CNN model, the MFCCs are further transformed into RGB images to leverage spatial feature extraction (introduced in Subection2.2.1). These features provide compact and perceptually relevant voice representations for machine learning models, by capturing key frequency components and temporal variations of voice signals. In contrast, Wav2Vec2.0-based models process raw audio signals directly through the framework's built-in processor. The processor automatically extracts task-related features from the raw waveform in an end-to-end manner, without the need for manual feature engineering. As a result, the model can more effectively adapt to diverse and complex speech tasks, and demonstrate stronger feature learning capabilities.

3.2 Model Selection

Selecting an appropriate model is critical to the success of voice disorder detection and classification tasks. Three models are evaluated in this study: MLP, 2D-CNN, and Wav2Vec2.0. The first two are used as baseline models for fine-tuning, providing simple and computationally efficient comparison points. In contrast, Wav2Vec2.0 is the primary focus of iur study, because it has an advanced architecture, supports direct processing of raw audio waveforms, and facilitates extensive exploration of model extensions and multimodal fusion strategies. This systematic comparison not only highlights the relative strengths and limitations of each model, but also lays the foundation for the detailed investigations presented in subsequent sections.

3.2.1 Compared models

MLP It was chosen as the benchmark model due to its simplicity and computational efficiency. It processes MFCC features directly, avoiding complex feature extraction. As shown in the left side of Fig. 3.3, the architecture consists of two dense layers with ReLU activations, followed by average pooling, and a final dense

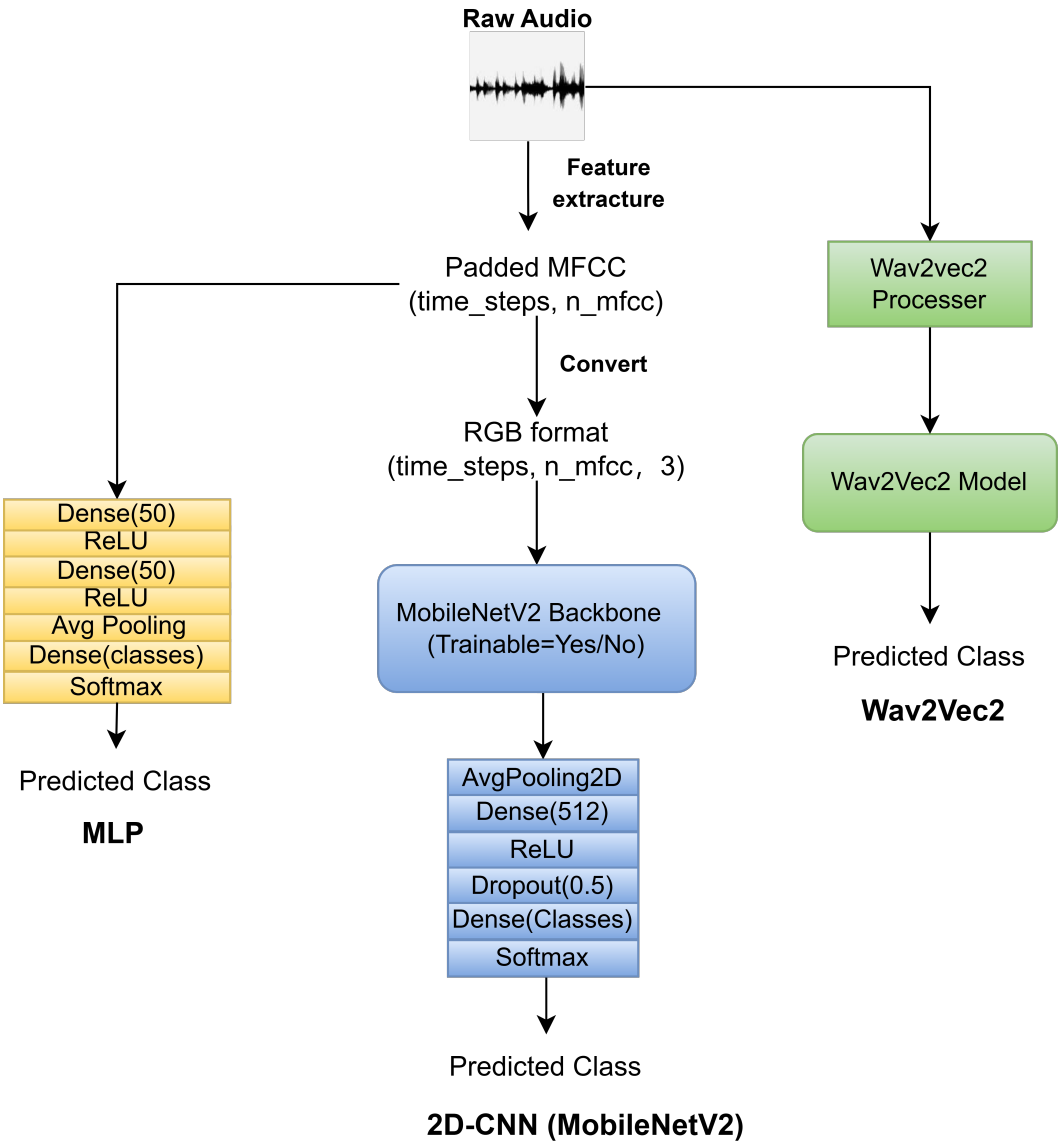


Figure 3.3: Comparison of three architectures: MLP, 2D-CNN (MobileNetV2), and Wav2Vec2.0. *Note: The figure illustrates the distinct processing pipelines of these architectures, from feature extraction to final prediction.*

layer with softmax activation for classification. The lightweight design of the MLP makes it a suitable starting point for benchmarking audio classification tasks.

The MLP has several advantages such as low computational effort, easy training, and is well suited for small datasets or resource-constrained scenarios. However, its limitations include the lack of ability to extract complex patterns from audio

features and heavy reliance on the quality of pre-computed MFCC features.

2D-CNN It employs a MobileNetV2 backbone, and serves as a stronger benchmark by leveraging convolutional operations to extract features from audio inputs. As shown in the middle of Figure 3.3, unlike MLP, 2D-CNN operates on MFCC features converted to RGB format, allowing it to utilize convolution operations for feature extraction. The MobileNetV2 backbone can be fine-tuned or frozen according to our choice, and the extracted features are further processed using average pooling, dense layers, and dropout for classification.

The advantage of 2D-CNN is that it can effectively capture temporal and frequency correlations, providing richer feature representations compared to MLP. In addition, using a pre-trained MobileNetV2 backbone can speed up training and enhance generalization capabilities. However, this model requires additional preprocessing to convert MFCC features to an image-like format, which incurs computational overhead. It is also less efficient than MLP in resource-constrained environments.

3.2.2 Wav2vec2.0

Wav2vec2.0 model is a state-of-the-art approach in the field of audio, chosen as the base model because of its simple architecture and ability to directly process raw audio waveforms. This end-to-end architecture does not require manual design or feature extraction, instead learning hierarchical feature representations through its Transformer-based design. As shown in Fig. 3.3, the model first normalizes the original waveform using a Wav2Vec2 processor, and then directly extracts feature representations through the Wav2Vec2 backbone. Finally, it is fine-tuned on a task-specific dataset to optimize performance (for more details on the internal mechanism of Wav2Vec2, see the Fig. 2.8 in the previous subsection 2.4.3).

Compared with MLP and 2D-CNN, Wav2Vec2 has significant advantages. It eliminates the reliance on hand-crafted features and, thanks to the pre-training strategy, performs particularly well in tasks with limited labeled data. In addition, due to its simple architecture, it is also easier to implement the multimodal fusion strategies introduced in subsequent sections 3.3, laying a solid foundation for further expansion work.

3.3 Fusion Strategies

This section presents our contributions to multimodal fusion strategies, focusing on voice pathology detection and multi-class classification tasks. By leveraging the Wav2Vec2 model, we explore robust feature extraction and fusion methods,

proposing and analyzing three-level fusion strategies: early, mid-level, and late fusion. These strategies systematically integrate features extracted from both modalities, allowing for a comprehensive evaluation, of their effectiveness in processing complementary information. Two voice-based modalities are employed in this study, each designed to capture unique and complementary characteristics of voice functionality:

1. Sentence reading recording (original feature x_1): captures dynamic voice patterns reflecting natural communication.
2. Sustained vowel pronunciation (original feature x_2): provides insights into voice stability, resonance, and sustained phonation capabilities.

The raw waveforms from these modalities are input into a pre-trained Wav2Vec2.0–based multimodal architecture f, which integrates features at various stages via the fusion strategy. The model outputs a probability distribution $\hat{y}_c$ over C classes, where C denotes the number of target classes. This unified framework supports both binary detection ($C = 2$) and multi-class classification ($C > 2$) tasks.

For all fusion strategies, the final predicted class label $\hat{y}$ is obtained by selecting the class with the highest predicted probability:

$$\hat{y} = \arg \max_{c \in \{1,\ldots,C\}} \hat{y}_c \tag{3.1}$$

3.3.1 Early Fusion

The early fusion strategy integrates the original features of the two modalities, into a unified input representation, enabling the model to directly learn cross-modal relationships. In order to standardize the input, and reduce the deviation caused by different sample lengths, all audio samples are first truncated or padded to a consistent length. The features of the two modalities are then combined in sequence, where modality x_1 is followed by modality x_2, and there is 1-second (s) of silence between them, which serves as a clear separator to ensure that the distinction between the modalities is maintained. This process generates a unified feature vector, as illustrated in Fig. 3.4, and can be formulated as:

$$x_{\text{early}} = [x_1; s; x_2] \tag{3.2}$$

where:

- x_1, x_2: Represent the features of the two modalities.
- s: Denotes the 1-second silence padding.
- $[;]$: Indicates the concatenation operation.

The concatenated feature vector x_{early} is input into the model f. Then the model computes a probability distribution $\hat{y}_c^{\text{early}}$ over C classes using the softmax function:

$$\hat{y}_c^{\text{early}} = \frac{\exp(f_c(x_{\text{early}}))}{\sum_{j=1}^{C} \exp(f_j(x_{\text{early}}))}, \quad c \in \{1, \ldots, C\} \tag{3.3}$$

where:

- $f_c(x_{\text{early}})$: Logit (unnormalized output) output for class c produced by the Wav2Vec2 model f with early fusion.
- $\exp(r)$: Exponential function, defined as e^r, ensuring that logits remain positive.
- $\hat{y}_c^{\text{early}}$: Probability for class c, obtained by normalizing logits via the softmax function.

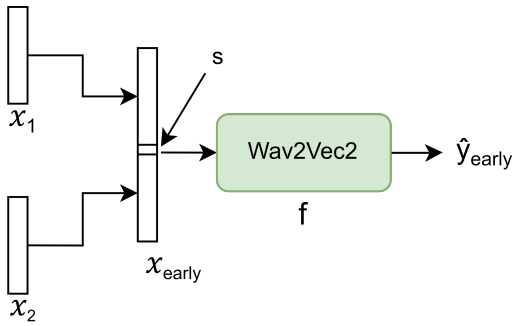


Figure 3.4: Diagram of the early-fusion

This method effectively preserves modality-specific characteristics, while leveraging the silence padding to maintain a clear distinction between modalities, making it a simple yet effective approach, for integrating complementary information from different data sources.

3.3.2 Mid-Level Fusion

Mid-level fusion incorporates modality-specific features at the intermediate stages of the Wav2Vec2 model. This process occurs after the initial CNN-based encoding, but before the transformer-based processing, allowing the features to be combined into a shared representation space. By leveraging this shared space, the model promotes richer and more meaningful interactions between modalities.

We initially considered four different approaches (For how to choose, please see the section. 4.2), but after preliminary evaluation, we chose two mid-level

fusion methods that worked best for our dataset: concatenated embeddings and cross-attention mechanisms. Details of the selection process are in the experiments section.

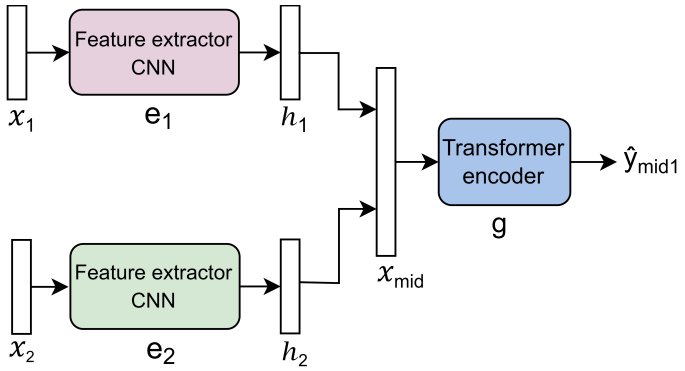


Figure 3.5: Mid-level fusion with concatenated embeddings

Concatenated Embeddings In this strategy, high-dimensional features h_1, h_2 from each modality are optionally normalized before concatenation, then are extracted using separate CNN-based encoders (e_1 and e_2), projected into a high-dimensional space, and then concatenated to form a unified representation x_{mid}. The overall process can be expressed as $f = e \circ g$, where $e = [e_1, e_2]$ represents the modality-specific encoders, and g is a shared transformer encoder. The concatenated embeddings are normalized and optionally passed through a dimensionality reduction layer to align with the transformer input size. The process is formalized as follows:

$$h_1 = e_1(x_1), \quad h_2 = e_2(x_2) \tag{3.4}$$

$$x_{\text{mid}} = [h_1; h_2] \tag{3.5}$$

where:

- h_1 and h_2: high-dimensional embeddings extracted from modalities x_1 and x_2 using a CNN extractor, respectively.
- x_{mid}: Concatenated feature embeddings from both modalities.

x_{mid} is then processed by g, which outputs logits for each class c:

$$\hat{y}_c^{\text{mid1}} = \frac{\exp(g_c(x_{\text{mid1}}))}{\sum_{j=1}^{C} \exp(g_j(x_{\text{mid1}}))}, \quad c \in \{1, \ldots, C\} \tag{3.6}$$

This method efficiently combines the features of both modalities while maintaining their modality-specific characteristics, as depicted in Fig. 3.5.

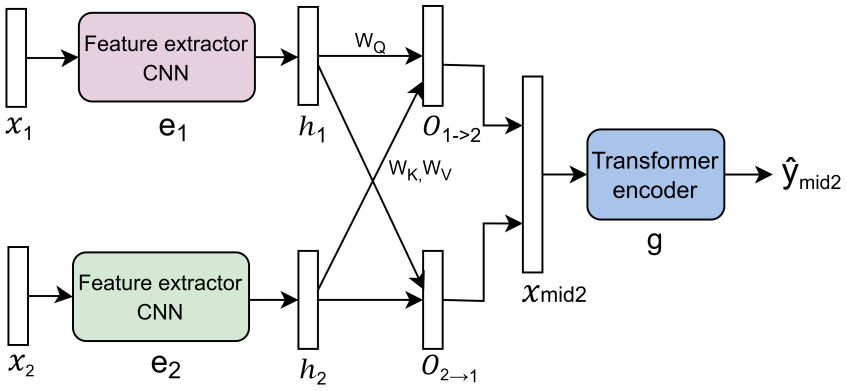


Figure 3.6: Mid-level fusion with cross-attention

Cross-Attention Mechanism The cross-attention mechanism dynamically identifies relationships between the two modalities by computing attention scores that emphasize critical features. Given feature matrices h_1 and h_2, we derive the Query (Q), Key (K), and Value (V) matrices as:

$$Q = h_1 W_Q, \quad K = h_2 W_K, \quad V = h_2 W_V \tag{3.7}$$

Here, W_Q, W_K, and W_V are learnable weight matrices for the query, key, and value, respectively. Next, we calculate the attention matrix A between the Query (Q) and the Key (K) by measuring their similarity, then normalized using softmax. The attention weight is used to perform a weighted sum of the Value V to generate output features O:

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \quad O = AV \tag{3.8}$$

where:

- A is the general attention matrix.
- d_k is the dimension of the key, $\sqrt{d_k}$ is the normalization factor used for scaling.

To capture bidirectional interactions, cross-attention is performed in both directions:

1. We use h_1 as the query and h_2 as the key and value to compute the attention.
2. We reverse the roles of the modalities and use h_2 as the query and h_1 as the key and value.

$$O_{1\to 2} = \text{CrossAttention}(h_1, h_2) \tag{3.9}$$
$$O_{2\to 1} = \text{CrossAttention}(h_2, h_1) \tag{3.10}$$

The outputs from both directions $O_{1\to 2}$ and $O_{2\to 1}$ are concatenated to form the fused representation:

$$x_{\text{mid2}} = [O_{1\to 2}; O_{2\to 1}] \tag{3.11}$$

This fused feature vector x_{mid2} is input to the transformer encoder g for classification, producing probability for each class as:

$$\hat{y}_c^{\text{mid2}} = \frac{\exp(g_c(x_{\text{mid2}}))}{\sum_{j=1}^{C} \exp(g_j(x_{\text{mid2}}))}, \quad c \in \{1, \ldots, C\} \tag{3.12}$$

As shown in Fig. 3.6, the bidirectional cross-attention mechanism enables the model to dynamically emphasize salient features from both modalities.

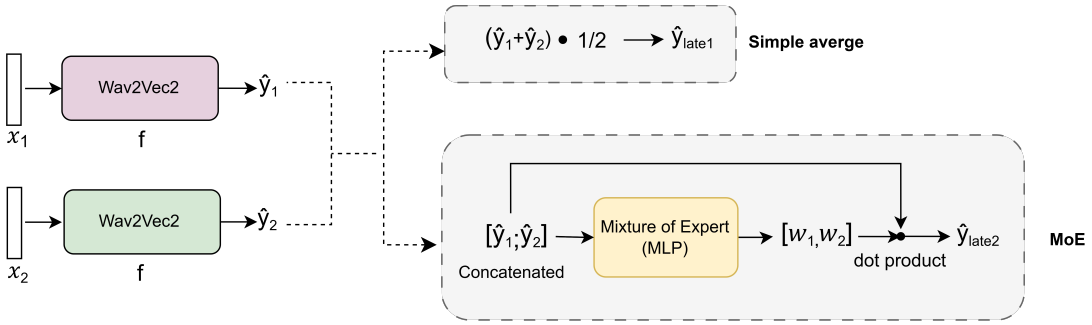


Figure 3.7: Two late fusion strategies

3.3.3 Late Fusion

Late fusion differs from other fusion strategies, allows the model of each modality to be optimized independently, in that its operation occurs at the decision stage. During the fusion process, the outputs of each modality (such as probability distributions or classification results) are combined to generate a unified prediction result. The primary advantage of late fusion lies in its flexibility and modular design. The model of each modality can be trained, replaced, or optimized independently without redesigning the entire system architecture.

In this study, we adopt two specific late fusion methods: simple averaging and shallow mixture of experts (MoE). Simple averaging method is an efficient and straightforward fusion method that assumes each modality contributes equally to the final prediction; while the MoE method dynamically adjusts the modality weights, to more flexibly adapt to the relative importance of different modalities to the task, thereby further optimizing the fusion performance.

Simple average In this method as in Fig. 3.7, in the case of two data modalities, the outputs of two models are combined, by calculating the average of the predicted probability distributions, $\hat{y}_{1,c}$ and $\hat{y}_{2,c}$, for all classes c, we made both models contribute equally to the final prediction. The combined output is calculated as follows:

$$\hat{y}_c^{\text{late1}} = \frac{1}{2}(\hat{y}_{1,c} + \hat{y}_{2,c}), \quad c \in \{1, \ldots, C\} \tag{3.13}$$

where:

- $\hat{y}_{1,c}$ and $\hat{y}_{2,c}$: Predicted probabilities for class c from the first and second models, respectively.
- $\hat{y}_c^{\text{late1}}$: Combined predicted probability for class c using the simple average strategy.

This method is both computationally efficient and simple, as it avoids the introduction of additional parameters, or the need for further training.

MoE As the second late fusion strategy, we introduce a shallow mixture of experts (MoE) method, to improve the overall system performance , by dynamically adjusting the output weights of two independent models. Compared with the simple averaging method, MoE can weight the contribution of each model prediction to the final output, thereby achieving more flexible and adaptive fusion.

As illustrated in Fig. 3.7, the MoE approach utilizes a simple MLP with a single hidden layer, to predict weights for combining the outputs of the two models. The input to the MLP is the probabilistic concatenation of the predicted probability distributions from the two models, denoted as x_{late2}:

$$x_{\text{late2}} = [\hat{y}_1; \hat{y}_2] \tag{3.14}$$

The output layer applies a softmax function to ensure that the sum of the predicted weights to 1:

$$w_q = \frac{\exp(z_q)}{\sum_{p=1}^{2} \exp(z_p)}, \quad z = \text{MLP}(x_{\text{late2}}), \quad q \in \{1, 2\} \tag{3.15}$$

In this equation:

- $x_{\text{late2}} = [\hat{y}_1; \hat{y}_2]$: The input to the MLP, which is the concatenation of the predicted probability distributions from the two models.
- $z = [z_1, z_2]$: The logits outputted by the MLP for each of the two models.

- w_q: The normalized weight for the q-th model, computed via the softmax function. The softmax ensures that $w_1 + w_2 = 1$, making the weights interpretable as probabilities. Higher values of z_q result in larger w_q, allowing the model to assign greater importance to one modality over the other.

During inference, the final prediction for each class c is computed, by weighting the predicted probabilities from the two models as follows:

$$\hat{y}_c^{\text{late2}} = w_1 \cdot \hat{y}_{1,c,\text{test}} + w_2 \cdot \hat{y}_{2,c,\text{test}}, \quad c \in \{1, \ldots, C\} \tag{3.16}$$

where:

- $[w_1, w_2]$: Weights predicted by the MLP based on $x_{\text{late2}} = [\hat{y}_1; \hat{y}_2]$ from the validation set.
- $\hat{y}_{1,c,\text{test}}$ and $\hat{y}_{2,c,\text{test}}$: Predicted probabilities for class c from the first and second models on the test set, respectively.
- $\hat{y}_c^{\text{late2}}$: Final predicted probability for class c after combining the outputs of both models using MoE.

Note that we use the non-adaptive weighting method, for the same data, the weight of each class is fixed. This method is relatively simple and easier to implement.

Chapter 4

Experiments and Results

In this chapter, we will present the overall design and specific settings of the experiment, which focus on two main tasks: voice disorder detection and classification. In the first part Section 4.1, we introduce the evaluation metrics, that will be used to measure the performance of the model. For the detection task, we take into account accuracy and macro F1 Score; for the classification task, in addition to these two indicators, we add the confusion matrix as well, in order to provide a more detailed analysis of model performance across different categories. In the second part Section. 4.2, we mainly introduce how the two ways of intermediate fusion are selected through specific experiments. Then in 4.3, we describe the computing environment in detail, including the hardware devices (GPU and CPU configuration) used to conduct the experiment, and the version information of the relevant software libraries. It also includes the specific configuration of the model used, as well as the parameters and specific steps used during the experiment. Finally, in Section 4.4 the experimental results are presented, providing a comprehensive comparison, and analysis for evaluating the performance of the model on different tasks and datasets. Through systematic experimental design, we can ensure that our model's ability in voice disorder tasks is strictly and comprehensively evaluated.

4.1 Evaluation metrics

To evaluate the performance of the model, we used the following key metrics:

Accuracy Accuracy measures the proportion of correctly predicted samples to the total number of samples, providing an overall assessment of the model's performance:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \tag{4.1}$$

While accuracy is a useful general metric, it may be less informative in the presence of class imbalance. Therefore, additional metrics are used to provide a more comprehensive evaluation.

Macro F1-Score Macro F1-Score is used to better handle class imbalance by evaluating performance between all classes equally. It is calculated as the average F1-Score of each class and incorporates precision and recall as core components.

To compute Precision and Recall, we first define the following terms:

- True Positives **(TP)**: Positive samples correctly predicted positive samples.
- False Positives **(FP)**: Negative samples incorrectly predicted as positive.
- False Negatives **(FN)**: Positive samples incorrectly predicted as negative.
- True Negatives (**TN)**: Negative samples correctly predicted as negative.

Based on these definitions, Precision and Recall are formulated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

The F1-Score is a metric that balances Precision and Recall, providing a single measure of a model's accuracy for a specific class. For a given class i, the F1-Score is defined as the harmonic mean of Precision and Recall:

$$F1_i = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

To evaluate performance across multiple classes, the Macro F1-Score is computed as the average F1-Score across all C classes:

$$\text{Macro F1-Score} = \frac{1}{C} \sum_{i=1}^{C} F1_i \tag{4.5}$$

Where:

- C: Total number of classes.
- $F1_i$: F1-Score of class ith, which depends on the class-specific Precision and Recall values.

Macro F1-Score ensures that the performance of minority classes is adequately reflected, making it particularly suitable for voice disorder detection and classification tasks.

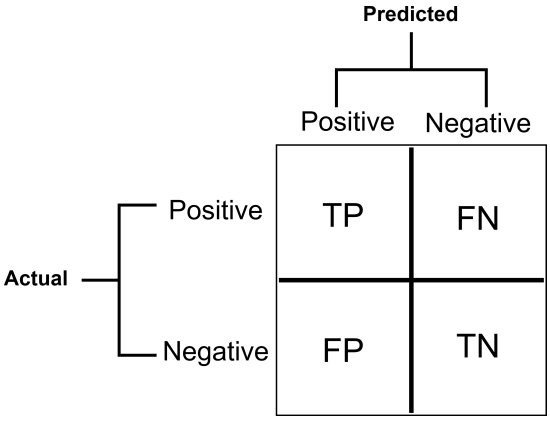


Figure 4.1: Confusion matrix

Confusion Matrix We further evaluated the model using a confusion matrix, shown in Figure 4.1, which provides a detailed view of the model's performance across all classes. Diagonal elements represent correct predictions, while non-diagonal elements represent incorrect classifications. For example, it can identify whether certain diseases are frequently misclassified as other categories, or whether underrepresented categories have a higher error rate. These findings can provide insights for targeted improvements, such as improving model architecture.

4.2 Mid-fusion method selection

When choosing three different levels of fusion strategies, early fusion, and late fusion use common methods. However, the mid-level fusion is more complicated, and there are many specific implementation methods. We need to choose the appropriate method to use on our dataset. In the selection process, we initially considered 4 specific methods, and finally selected two of them for application in our experiments based on the effect and cost. This section will describe how we selected them.

Based on the concept of mid-level fusion, we should integrate high-dimensional features of different modalities to enhance classification performance. Wave2Vec2.0 consists of a CNN feature extractor followed by a Transformer encoder. Given this structure, we considered the following four fusion methods:

1. Pre-class fusion: For the two modalities (CS, SV), feature extraction is performed through two pre-trained Wave2Vec2.0 models, and features are concatenated after the Transformer encoder outputs, and finally input into the classifier for decision-making.

2. Concatenated embeddings: The features of the two modalities from CNN are directly concatenated, and then input into a shared Transformer encoder for further modeling.

3. Cross-attention: Similar to the previous one, but not directly concatenated, using bidirectional multi-head attention to allow CS and SV modalities, to interact with each other to enhance cross-modal information sharing, then concatenate the two attention matrices and put them into a shared Transformer encoder.

4. Co-attention: After CNN extracts features, each modality interacts with each other, through a dedicated 3-layer cross-attention Transformer (each layer contains 2 multi-head attention mechanisms). The final encoded features are then averaged using average pooling, and passed into the classifier for decision-making.

4.2.1 Synthetic dataset construction

In order to select a suitable model, we constructed a synthetic dataset based on the original dataset IPV. By artificially introducing noise, we simulated various noise interference situations that may be encountered in real environments, and preliminarily evaluated the adaptability and robustness of each fusion model under noise conditions.

The specific construction method is to randomly select individuals in IPV, and their CS and SV are processed in 1, 2, 3, or 4 ways according to the following rules, and then put them into the synthetic dataset after processing. This cycle repeats from 1 to 4 until all samples in IPV are used up:

1. **(CS, SV)**: Both modes are clean;

2. **(CS+noise, SV)**: Only CS has noise;

3. **(CS, SV+noise)**: Only SV has noise;

4. **(CS+noise, SV+noise)**: Both modes have noise.

Each pair in the ( ) corresponds to the same individual. Noise means adding stronger 0.1 Gaussian noise, and the number of samples of each type in the final synthetic dataset is basically the same.

4.2.2 Experimental protocol

We evaluated the effectiveness of these different mid-level fusion strategies, on a 4-class classification task using a synthetic dataset. The goal is to predict whether

there is noise in CS or SV. Each sample belongs to one of the following four categories: no noise in both modalities; only CS has noise; only SV has noise; both have noise. For two modalities, they are fed as input to Wav2Vec2.0, and based on different mid-level fusion strategies, the model predicts whether there is noise in a given modality.

Since our goal is to compare ensemble methods rather than optimize hyper-parameters, we ensure fair evaluation by using the same training settings for all ensemble strategies. We use the AdamW optimizer with a learning rate of 1e-5 and a batch size of 8. Each model is trained for a maximum of 30 epochs with early stopping based on validation loss to avoid overfitting.

To evaluate the models, we use the confusion matrix because it allows us to analyze the error distribution, see how well each fusion strategy distinguishes clean and noisy conditions, and intuitively compare the reliability of different fusion strategies.

4.2.3 Results & Analysis

According to the confusion matrix in Figure 4.2, let's analyze their results:

- Model (1) (Top Left): This model appears to be overly sensitive to noise features, which may lead to overfitting of noise patterns rather than effectively distinguishing between clean and noisy conditions.
- Model (2) (Top Right): This is the best-performing model, achieving the highest classification accuracy across all categories and demonstrating strong robustness to noise.
- Model (3) (Bottom Left): Although some miss classifications occur in the noisy CS category, the overall performance remains good.
- Model (4) (Bottom Right): This model also performs poorly, however, unlike Model (1), it struggles to distinguish whether SV contains noise but retains strong identification ability for CS.

Due to the poor performance of Models (1) and (4) in distinguishing noise, we will focus on improving the better-performing models (2) and (3), as described in Section 3.3.

4.3 Experimental design

This section introduces the experimental setup, including the computing environment, baseline models, training configurations, and experiment settings used in our study.

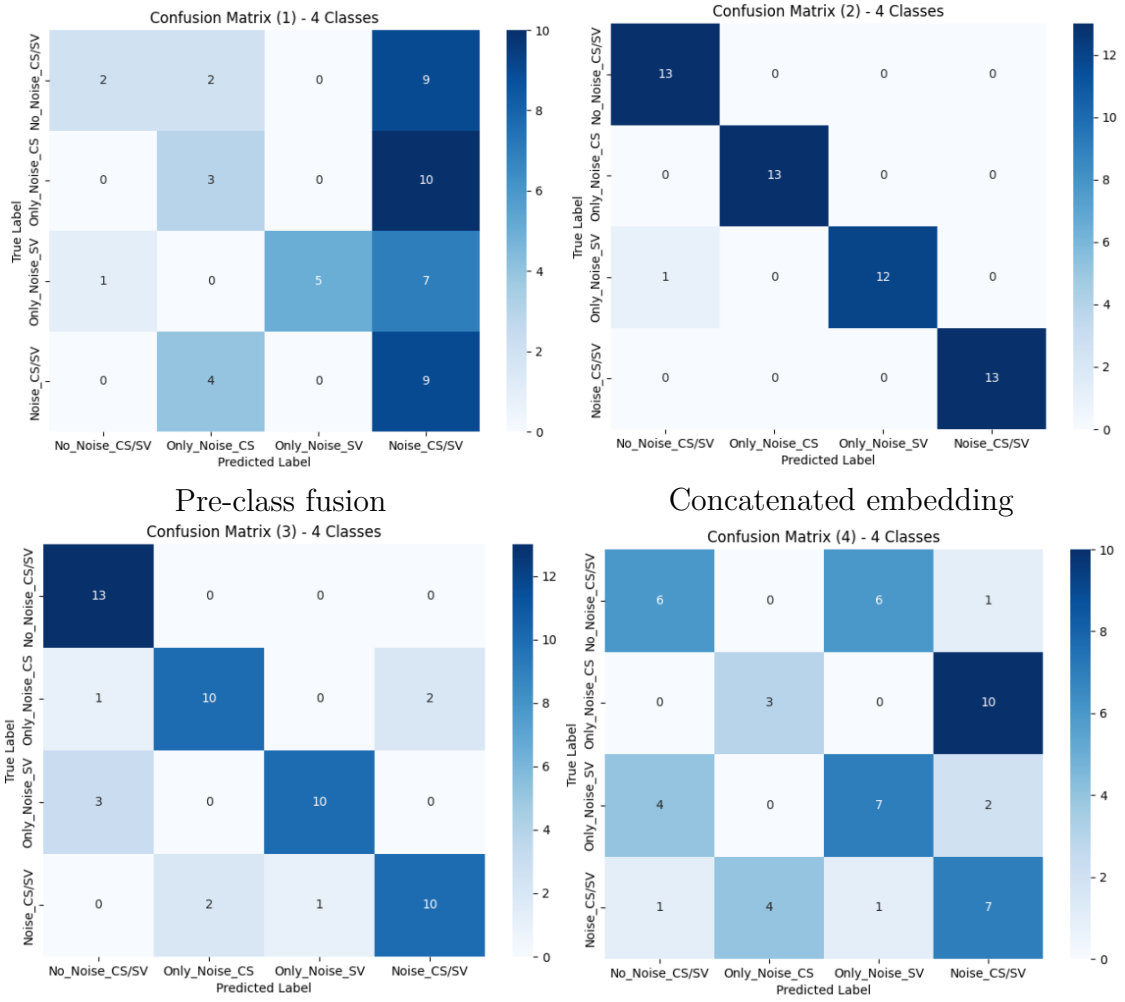


Figure 4.2: Confusion matrix of synthetic data fused in four intermediates

We first outline the hardware and software configurations used for model training and evaluation in Subsection 4.3.1, followed by the introduction of the baseline models used for comparison, detailing their architectures and hyperparameter settings in Subsection 4.3.2. Furthermore we can see Subsection 4.3.3, which describes the training procedures for voice pathology detection and multi-class classification tasks, including learning rate scheduling, optimizer selection, loss functions, and strategies to prevent overfitting. Finally, we present the settings for all experimental strategies, focusing on explaining the specific architecture hidden layers, as well as the transfer learning settings from 8 categories to 6 categories, and the specific practices of domain adaptation on new datasets in Subsection 4.3.4.

4.3.1 Experimental environment

All our experiments were run in a virtual environment [1], mainly using an NVIDIA Tesla P100 GPU for model training and inference. This GPU is equipped with 16 GB of memory to effectively handle large-scale matrix calculations in deep learning tasks. The environment also includes an Intel(R) Xeon(R) CPU @ 2.00GHz processor with 4 logical cores and 38.5 MiB of L3 cache. In addition, there is 29 GB of RAM available for data loading and processing, and 20 GB of temporary storage for datasets and intermediate results.

The experiments used Python version 3.10.14 and the PyTorch framework (v2.4.0) for model training and evaluation. Other libraries used include NumPy (v1.26.4) for numerical calculations, Pandas (v2.2.2) for data processing, and Scikit-learn (v1.2.2) for performance evaluation and metric calculation. Since a GPU is used, its CUDA version is 12.3 and cuDNN version is 90000, which ensures efficient execution of computationally intensive tasks.

And to ensure reproducibility, we apply a fixed random seed in all stages of training and evaluation. The random number generators of Python, NumPy, and PyTorch random number generators (CPU and GPU operations) are configured with the same seed value of 42. This unified setting can reduce randomness in the experimental results, thus ensuring that experiments under the same conditions can be reliably reproduced.

4.3.2 Baseline Model

To make the comparisons of our approach clearer, this section is based on the baseline model architectures, introduced in the methodology section (Section 3.2.1) and the architecture diagram (Fig. 3.3). We will explain in detail the specific settings and considerations for the baseline model during training. And the benchmark experiments are mainly for unimodal data, with slightly different configurations for detection and classification tasks.

MLP For this model, 40-dimensional MFCC features are used as input, a two-layer fully connected network is designed with 50 hidden units in each layer, and ReLU activation function to extract high-dimensional features. Then, they are aggregated by using a global average pooling layer and a softmax output layer is used to complete binary or multi-classification tasks.

[1]We gratefully acknowledge the computational resources provided by Kaggle (`https://www.kaggle.com/`) for this research. We also appreciate the early-stage support from HPC@Polito (`http://www.hpc.polito.it`).

The optimizer used is Adam, the learning rate is 0.01, and the batch size is 16. Moreover, the loss function is selected depending on the task: binary cross-entropy is used for binary classification tasks, and categorical cross-entropy is used for multi-classification tasks.

Model training is set with a maximum of 50 epochs, and early stopping is implemented (if the performance is not improved with 10 epochs) to avoid overfitting.

2D-CNN In the 2D-CNN model, the 40-dimensional MFCC features are first converted to 3-channel RGB images suitable for CNN input. Then, the ImageNet pre-trained MobileNetV2 model is used as the basis, the top classification layer is removed, and a GlobalAveragePooling2D and a fully connected layer containing 512 units are added. In order to enhance the generalization, the dropout operation with 0.5 is added to the top network, and then the detection or classification task is completed through the Softmax layer.

We used two fine-tuning strategies in model training: full fine-tuning and head-only fine-tuning. In full fine-tuning, all layers of MobileNetV2 are involved in training to optimize the overall performance; while in head-only fine-tuning, only the newly added classification head is updated, and the pre-trained feature extraction layer is frozen to retain the common features learned from ImageNet. Regardless of the strategy adopted, the training hyperparameters are consistent with the MLP model, including optimizer, learning rate, batch size, and early stopping strategy.

4.3.3 Training Configuration

To ensure the controllability and repeatability of the training process, this subsection introduces key hyperparameters and optimization strategies, including optimizers, learning rate scheduling, loss functions, early stopping mechanisms, etc. We also adjusted the class imbalance problem of multi-classification tasks. In addition, we explain the reason for using the Hugging Face Trainer framework and why a custom PyTorch training loop is needed.

The experiments were all completed within 50 epochs, and a fixed random seed was used to ensure the reproducibility of the results. We used the common AdamW optimizer (weight decay = 0.01), and optimized using the linear learning rate scheduler. This scheduler has no warm-up and decays the learning rate linearly with the training steps. The initial learning rate was optimized by manual adjustment, with 6e-6 used for the cross-attention method involving mid-level fusion, and 1e-5 used for all other methods (unimodal, augment only, early fusion, concatenated embeddings of the mid fusion, late fusion). In multi-classification tasks, due to the severe imbalance in the distribution of categories, we use weighted cross-entropy as the loss function, and calculate the weights according to the

category distribution in the training dataset to reduce the model's bias towards the majority class. For the detection task, the ratio of healthy to pathological in the IPV dataset is about 1:2, and the category imbalance is not particularly severe, so using a normal cross-entropy loss is also sufficient.

By default, we use the Trainer [2]training framework provided by Hugging Face to simplify the code, and take advantage of its optimization capabilities for the standard Transformer structure. Trainer is suitable for single-modal training, data augmentation, early fusion, and late fusion, and can automatically manage tasks such as data loading, gradient updates, and learning rate scheduling. However, once mid-level fusion is involved, the architecture of Wav2Vec2.0 is modified to adopt a dual-stream Wav2Vec2.0 + shared Transformer structure, which makes Trainer unable to adapt directly. If it is forced to use it, a lot of custom modifications to Trainer are required, which is too costly. Therefore, in the mid-level fusion, we manually wrote a PyTorch training loop, including custom data loading, model forward propagation (through their own Wav2Vec2.0, and then into the shared Transformer structure), loss calculation, backpropagation, gradient update, and early stopping strategy.

The default batch size is set to 8, but in the mid-Level fusion task, due to the large memory usage of the model, we adjust the batch size to 4 and use gradient accumulation twice, to make it equivalent to a batch size of 8 to ensure the uniformity of the training method. To improve training efficiency and prevent overfitting, we use an early stopping strategy. When the validation set performance does not improve significantly within 10 epochs, the training is terminated early. More experimental details can be found in the GitHub repository of this article [3].

4.3.4 Experimental Settings

This subsection provides the implementation details of our experiments, focusing on parameters unrelated to model training.

These methods are based on the pre-trained Wav2Vec2.0 model (trained on the LibriSpeech 960-hour dataset), mainly including unimodal fine-tuning, data augmentation, and multimodal fusion strategies. We first apply these methods to binary classification tasks, and then extend them to multi-class classification. In addition, we fine-tune the model on previously unseen subsets of data, significantly improving model generalization. Additionally, we transition from an 8-class to a 6-class classification task, to investigate whether fine-tuning on a pre-trained binary classification model yields better results than directly training on a multi-class.

[2]For details in `https://huggingface.co/docs/transformers/en/main_classes/trainer`

[3]`https://github.com/qingqingkk/qingqingkk-Thesis.git`

Data augmentation Our data augmentation pipeline is mainly based on three time-domain augmentation techniques, where parameters are randomly sampled from a uniform distribution to ensure training data diversity. The parameter selection is based on the physical properties of human voice signals, to preserve natural variations while preventing excessive distortion.

- Gaussian white noise is added to the waveform, and the noise amplitude is randomly selected in $\mathcal{U}(0.001, 0.015)$ [4], ensuring that the noise level is neither too weak to be effective, nor too strong to reduce speech intelligibility, thereby enhancing the model's robustness to different noise environments.
- Time stretching is applied by randomly adjusting the playback speed within $\mathcal{U}(0.8, 1.25)$, which is consistent with natural speech rate variations, while avoiding excessive time distortion.
- Pitch shifting is performed by randomly changing the pitch within $\mathcal{U}(-4, 4)$ semitones to simulate different vocal characteristics of different speakers.

In addition to individual augmentations, we also adopt a combined augmentation strategy, which is a sequential combination of noise injection, time stretching, and pitch shifting on selected samples. This approach enhances the model's ability to generalize across different acoustic conditions, and adapt to speakers with different voice characteristics.

Fusion strategies Early fusion is performed by directly concatenating the raw audio of CS, SV and adding 1 second of silence after the total length of the audio (38 seconds) to avoid feature loss. This concatenation is performed on the same individual. The concatenated audio signals are uniformly processed in a Wav2Vec2.0 processor to ensure consistency in feature extraction.

Mid-level fusion is based on two fine-tuned Wav2Vec2.0 models, initialized with pre-trained Wav2Vec2 parameters. In practice, we use unimodal models that have been pre-trained on our dataset, and freeze the CNN encoder, allowing only the shared Transformer layers to be fine-tuned. This approach significantly reduces training costs. At this stage, the first mid-fusion directly concatenates the features extracted from CS and SV, and the second one implements feature interaction through a bidirectional cross-modal attention mechanism. The number of attention heads is set to 4.

Late fusion utilizes the fine-tuned CS and SV models to generate the final classification result, by combining the probabilities of the two modes, either through

[4] $\mathcal{U}(a, b)$ represents a uniform distribution, where values are sampled from the continuous interval $[a, b]$.

simple averaging or shallow part (MLP with 10 hidden nodes), with mode weighting determined based on the probabilities of the training and validation sets.

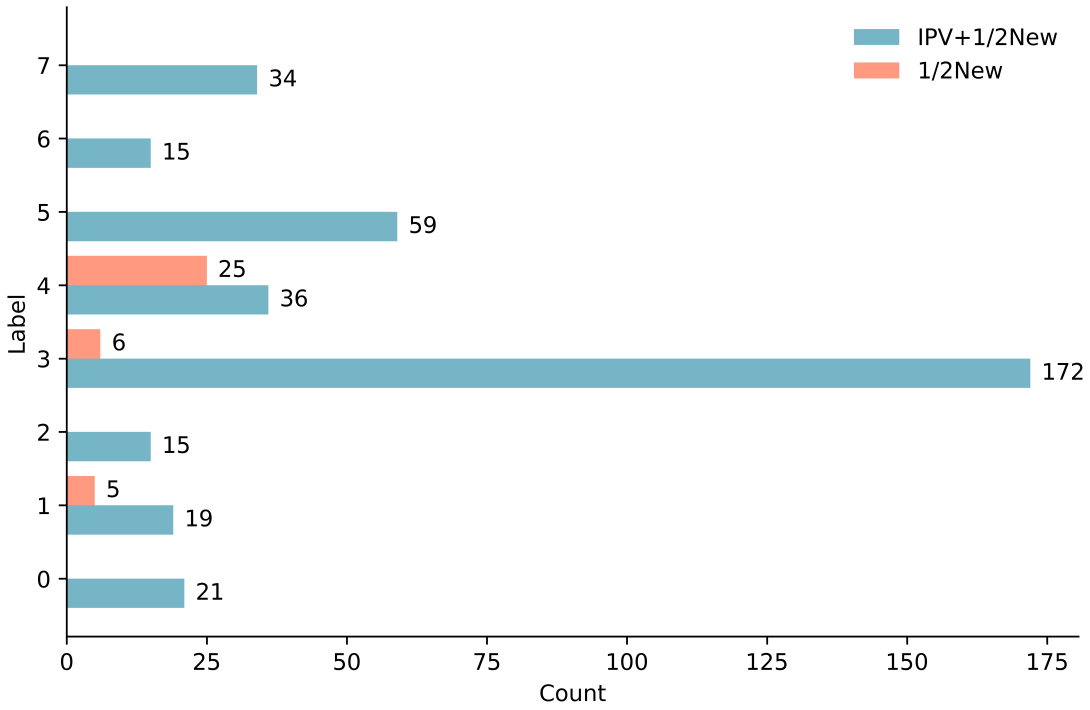


Figure 4.3: Label distribution comparison between IPV+1/2New and 1/2New (8 classes)

Generalization to unseen data We also conducted some interesting experiments, to evaluate the generalization ability of the 8-class classification model, we first evaluate the unimodal method on the new dataset, which contains 3 categories with environmental noise, while there are 8 categories in IPV. This distribution difference poses a challenge for domain adaptation. To address this problem, we consider using an adaptive approach based on fine-tuning the model. Randomly split half of the new dataset by category, select 4, 6, and 25 samples from labels 1, 3, and 4, respectively, and integrate them into the IPV dataset before fine-tuning Wav2Vec2.0. The label distribution of the integrated IPV dataset and the remaining half of the new dataset is shown in Fig. 4.3. Finally, we evaluate the fine-tuned model on the remaining unseen half of the new dataset, to check its adaptation performance in noisy environments.

In addition, we use the same method to conduct generalization experiments in the binary classification task. However, since the ratio of healthy and diseased

categories in the new dataset, which is similar to the IPV dataset (such as the number of samples in Table 3.1), we directly randomly select half of the healthy and diseased categories in the new dataset, and add them to IPV for fine-tuning.

4.4 Results

Next, we present experimental results, first comparing unimodal and multimodal methods for binary and multi-class classification. Then, we analyze the performance improvement, and generalization achieved by our data augmentation method. Finally, we explore the impact of fine-tuning adaptation methods on model performance.

4.4.1 Unimodal vs. Multimodal

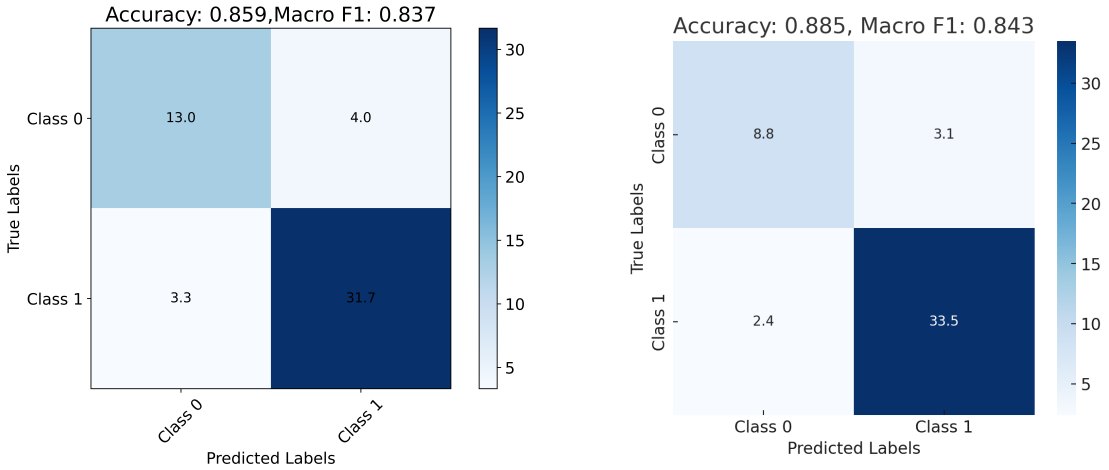


Figure 4.4: Comparison of average confusion matrices for Wav2Vec2 single-modality (CS, left), and multimodal cross-attention (right) in the detection task.

Voice pathological detection In Table. 4.1, we can see that in the detection task, Wav2Vec2.0 performs significantly better than other unimodal models, with an accuracy improvement of 0.058 (CS) and 0.077 (SV) over MLP, and 0.192 (CS) and 0.154 (SV) over CNN (training all layers). And its macro F1 score is also the highest among unimodal methods, confirming the powerful feature extraction ability of self-supervised learning. This is why we prefer Wav2Vec2.0 as the base model.

When moving from an unimodal model to a multimodal model, the multimodal fusion strategy further improves the performance of Wav2Vec2.0 without data

Modality	Method	Accuracy		Macro F1	
		CS	**SV**	**CS**	**SV**
Single	MLP	0.801 ± 0.011	0.750 ± 0.058	0.767 ± 0.022	0.686 ± 0.053
	2D-CNN (Train all layers)	0.667 ± 0.011	0.673 ± 0.000	0.400 ± 0.004	0.402 ± 0.000
	2D-CNN (Fine-tune classify head)	0.789 ± 0.019	0.782 ± 0.048	0.765 ± 0.021	0.723 ± 0.063
	Wav2Vec2.0	0.859 ± 0.029	0.827 ± 0.000	0.837 ± 0.038	0.793 ± 0.000
Multi	Early Fusion	0.859 ± 0.011		0.829 ± 0.016	
	Mid (Concatenated Embeddings)	0.878 ± 0.011		0.838 ± 0.014	
	Mid (Cross Attention)	**0.885±0.000**		0.843 ± 0.005	
	Late (Simple Average)	0.853 ± 0.022		0.824 ± 0.027	
	Late (MoE)	0.872 ± 0.011		**0.857±0.012**	

Table 4.1: Performance comparison for detection tasks on IPV.

augmentation. Mid-Level cross attention achieves the highest accuracy at 0.885, improving CS by 0.026 and SV by 0.058 compared to Wav2Vec2.0. Meanwhile, MoE achieves the best macro F1 at 0.857, improving CS by 0.02 and SV by 0.064. These results highlight that cross-modal interactions enhance deeper feature learning, particularly benefiting SV, which sees a larger improvement compared to CS. However, early fusion (0.859 accuracy) does not improve over unimodal Wav2Vec2.0 in either CS or SV, indicating that simply merging features at an early stage, may not fully utilize the complementary nature of the modalities. The mid-level concatenation approach provides moderate improvements (CS: +0.019, SV: +0.051), while late fusion (simple average) and MoE improve macro F1 more significantly than accuracy, suggesting that these methods are particularly useful for balancing class distributions rather than improving overall correctness. The confusion matrix 4.4 shows that, Wav2Vec2.0 has achieved stable binary classification performance under CS unimodality (left), while the cross-attention method (right), further improves the balance of classification. This once again proves that in binary classification tasks, it is reasonable not to apply weighting during training. The model achieves balanced performance without the need for distribution adjustments.

When testing our baseline and fused models on new datasets, we can see some different trends, as shown in Fig. 4.2. Mid-level fusion using concatenated embeddings is the best-performing method (accuracy: 0.741, macro F1: 0.501), while cross-attention fusion does not bring additional gains. Early fusion fails completely (accuracy: 0.278, macro F1: 0.219), but late fusion (MoE) achieves better generalization than simple averaging (accuracy: 0.735, macro F1: 0.444), consistent with its effectiveness in balancing class distribution.

In summary, although Wav2Vec2.0 is a strong end-to-end model, fusion (especially mid-level cross-attention and MoE) further improves its performance on our dataset, demonstrating the power of multimodal learning for voice disorder

Modality	Method	Accuracy		Macro F1	
		CS	**SV**	**CS**	**SV**
Single	MLP	0.716 ± 0.012	0.719 ± 0.014	0.576 ± 0.029	0.543 ± 0.030
	2D-CNN (Train all layers)	0.719 ± 0.014	0.747 ± 0.011	0.428 ± 0.012	0.562 ± 0.027
	2D-CNN (Fine-tune classify head)	0.725 ± 0.005	0.756 ± 0.005	0.431 ± 0.017	0.543 ± 0.022
	Wav2Vec2	0.735 ± 0.005	0.735 ± 0.014	0.444 ± 0.037	0.529 ± 0.026
Multi	Early Fusion	0.278 ± 0.000		0.219 ± 0.018	
	Mid (Concatenated Embeddings)	$\mathbf{0.741 \pm 0.000}$		$\mathbf{0.501 \pm 0.039}$	
	Mid (Cross Attention)	$\mathbf{0.741 \pm 0.009}$		0.475 ± 0.048	
	Late (Simple Average)	0.728 ± 0.005		0.432 ± 0.019	
	Late (MoE)	0.735 ± 0.005		0.444 ± 0.037	

Table 4.2: Performance comparison for detection tasks on the New dataset.

detection.

Modality	Method	Accuracy		Macro F1	
		CS	**SV**	**CS**	**SV**
Single	MLP	0.578 ± 0.017	0.490 ± 0.017	0.167 ± 0.058	0.193 ± 0.085
	2D-CNN (Train all layers)	0.520 ± 0.017	0.510 ± 0.017	0.131 ± 0.017	0.096 ± 0.022
	2D-CNN (Fine-tune classify head)	0.549 ± 0.017	0.549 ± 0.017	0.144 ± 0.010	0.188 ± 0.043
	Wav2Vec2.0	0.637 ± 0.068	0.549 ± 0.045	0.410 ± 0.110	0.175 ± 0.071
Multi	Early Fusion	$\mathbf{0.686 \pm 0.017}$		0.303 ± 0.040	
	Mid (Concatenated Embeddings)	0.598 ± 0.045		0.203 ± 0.039	
	Mid (Cross Attention)	0.608 ± 0.061		0.258 ± 0.185	
	Late (Simple Average)	0.618 ± 0.059		$\mathbf{0.396 \pm 0.100}$	
	Late (MoE)	0.598 ± 0.034		0.372 ± 0.093	

Table 4.3: Performance comparison of 8-class classification tasks.

Voice disorder 8-class classification The classification task is significantly more challenging than the detection task, with all methods performing at a lower level (Tab. 4.3). Not surprisingly, Wav2Vec2.0 again leads the unimodal methods, achieving 0.637 accuracy on CS and 0.549 on SV, while CNN (train all layers) performs the worst (0.520 accuracy on CS and 0.510 on SV). The macro F1 scores further highlight the class imbalance issue, with CNN (train all layers) performing poorly (0.131 on CS, 0.096 on SV) while Wav2Vec2.0 achieves a much higher 0.410 F1 on CS, suggesting that self-supervised learning provides better feature generalization. However, since labels 3 and 5 account for 67% of the total IPV, the model has learned the features of these two categories more fully, resulting in a clear bias in the prediction results towards them.

Confusion matrix analysis (Fig. 4.5) confirms this bias: in the CS modality (IPV dataset), class 3 has the highest correct classification rate (43.1%), while

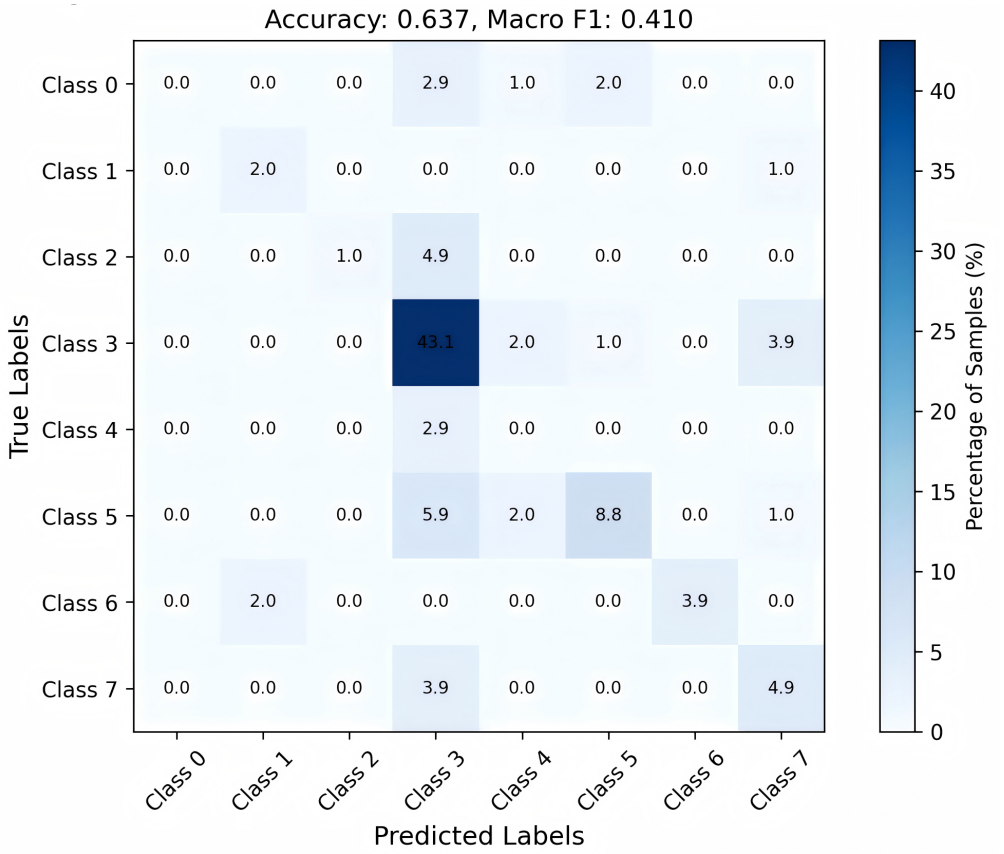


Figure 4.5: Average confusion matrix on IPV for Wav2Vec2.0 single-modality (CS) in the 8-class classification task.

class 5 is also well-classified (8.8%). However, substantial confusion exists between class 4 and class 5, likely due to their similar acoustic features. On the New dataset, performance drops sharply (accuracy: 0.216, macro F1: 0.116), with class 4 misclassified as class 5 in 36.6% of cases, illustrating the impact of training data imbalance on generalization.

Unlike detection (binary classification), multimodal fusion does not always provide a significant improvement in 8-class classification. Early fusion achieves the highest accuracy (0.686), improving Wav2Vec2.0 by 0.049 (CS) and 0.137 (SV), but its macro F1 drops by 0.107 on CS, indicating a persistent class imbalance. Surprisingly, mid (concatenated) and late fusion (MoE) do not improve overall accuracy over Wav2Vec2.0 and even decrease CS performance.

4.4.2 Data augmentation

This subsection shows the effects of different augmentation levels on binary and multi-classification tasks. First, we analyze the effects of augmentation on unimodal (CS and SV) and multimodal models, and compare their performance differences under different augmentation strengths. Then we explore the impact of data augmentation on the generalization ability of new datasets, and evaluate the effectiveness of different augmentation strategies in improving model adaptability.

Methods	Modality	Augmentation Strength	IPV		New	
			Accuracy	F1 Macro	Accuracy	F1 Macro
wav2vec2	CS	0	0.859 ± 0.029	0.837 ± 0.038	0.735 ± 0.005	0.444 ± 0.037
		30%	0.853 ± 0.011	0.838 ± 0.019	0.744 ± 0.033	0.506 ± 0.102
		50%	**0.885 ± 0.000**	**0.862 ± 0.011**	0.753 ± 0.028	0.508 ± 0.082
		70%	0.853 ± 0.029	0.833 ± 0.029	**0.799 ± 0.014**	**0.677 ± 0.037**
	SV	0	0.827 ± 0.000	0.793 ± 0.000	0.735 ± 0.014	0.529 ± 0.026
		30%	**0.859 ± 0.011**	0.801 ± 0.079	0.759 ± 0.037	0.610 ± 0.087
		50%	0.849 ± 0.024	**0.833 ± 0.027**	**0.787 ± 0.019**	**0.676 ± 0.039**
		70%	0.821 ± 0.011	0.787 ± 0.023	0.747 ± 0.014	0.572 ± 0.052
Mid (cross attention)	Multi	0	0.885 ± 0.000	0.843 ± 0.004	0.741 ± 0.009	0.475 ± 0.048
		30%	**0.897 ± 0.011**	**0.882 ± 0.016**	0.738 ± 0.005	0.499 ± 0.048
		50%	0.891 ± 0.011	0.875 ± 0.018	**0.759 ± 0.032**	**0.576 ± 0.112**
		70%	0.846 ± 0.033	0.818 ± 0.050	0.741 ± 0.019	0.532 ± 0.050

Table 4.4: Evaluating augmented models on IPV and New datasets for the detection task.

Augmentation for detection and classification The intensity of data augmentation has a significant impact on both binary and multi-classification tasks on the IPV dataset. The following describes the performance comparison of the model with and without augmentation.

In binary classification (left side of the Table 4.4), for CS, 50% augmentation produces the highest accuracy 0.885(+0.026) and macro F1 0.862(+0.025), however, at 70% augmentation, the accuracy drops to 0.853 and macro F1 drops to 0.833; for SV, the best performance occurs at 30% augmentation, achieving the highest accuracy 0.859(+0.032), indicating that SV benefits from lighter augmentation. At 50%, accuracy increases to 0.849, but macro F1 to the peak of 0.833(+0.04), however, at 70%, both accuracy and macro F1 drop. These results suggest that CS benefits most from moderate augmentation (50%), while SV achieves the highest accuracy at lower levels (30%). We also experimented with the best multimodal model, the cross-attention method combined with data augmentation, which achieved the highest improvement at 30% augmentation. The accuracy improved by 0.012, reaching 0.897, and the macro F1 increased by 0.039, reaching 0.882.

Table. 4.5 shows the effects of different augmentation strengths on the 8-class

Modality	Augmentation Strength	Accuracy	F1 Macro
CS	0	0.637 ± 0.068	**0.410±0.110**
	30%	0.676 ± 0.029	0.355 ± 0.148
	50%	**0.686±0.017**	0.386 ± 0.051
	70%	0.598 ± 0.034	0.240 ± 0.135
SV	0	0.549 ± 0.045	0.175 ± 0.071
	30%	**0.608±0.017**	**0.252±0.080**
	50%	0.598 ± 0.017	0.226 ± 0.058
	70%	0.569 ± 0.017	0.212 ± 0.063

Table 4.5: Effect of different augmentation strengths on 8-class classification performance in the IPV dataset.

classification performance of CS and SV modalities in the IPV dataset. Similar to binary classification, CS and SV achieve the best performance at different augmentation strengths, with CS benefiting the most at 50% and SV peaking at 30%. However, the effect of augmentation on multi-class classification is more pronounced. For CS, 50% augmentation yields the highest accuracy 0.686(+0.049), but unlike binary classification, macro F1 does not improve proportionally (0.386, -0.024), suggesting that while augmentation helps overall classification, it does not necessarily improve class balance. In contrast, 70% augmentation results in a large drop in accuracy (-0.039) and an even larger drop in macro F1 (-0.17). For SV, the best performance occurs at 30% augmentation, with an accuracy of 0.608 (+0.059) and macro F1 of 0.252 (+0.077), however, unlike binary classification, 50% or 70% augmentation is still effective for SV, but here accuracy compares with 30% to drop at 50% and drops further at 70%.

These results show that CS benefits the most from moderate augmentation, while SV is better suited at a lower augmentation level. Although augmentation can improve performance, excessive augmentation can introduce negative effects.

Impact of data augmentation on generalization to New data Data augmentation plays a crucial role in improving the generalization ability of new datasets, especially when dealing with distribution changes. Among them, CS achieves the best generalization at 70% augmentation, while SV or multimodal benefits the most at 50% augmentation.

As shown on the right side of Table 4.4, in the detection task, the appropriate use of augmentation strategies significantly improves the accuracy of new datasets. For CS, 70% augmentation achieves the highest accuracy (0.799, +0.064) and macro F1 (0.677, +0.233) on the new dataset, indicating that strong augmentation significantly improves the generalization ability on unseen data, and the higher

variability of training data helps the model adapt to domain changes, thereby reducing overfitting on the IPV dataset. However, in the IPV dataset itself, 70% augmentation has a negative impact on accuracy, confirming that the model sacrifices in-domain accuracy in exchange for better generalization ability. For SV, the best generalization occurs at 50% augmentation, with an accuracy of 0.787 (+0.052) and a macro F1 of 0.676 (+0.14). In the multimodal setting, the best generalization also occurs at 50% augmentation, but the results show that the generalization performance of multimodal models is weaker compared to unimodal models. Therefore, applying a dataset-specific or model-specific augmentation strength is critical to reach the best generalization performance.

4.4.3 The impact of fine-tuning strategies

This subsection compares, a model trained exclusively on the IPV dataset, to a model fine-tuned using a combination of samples from the IPV and new datasets. The results highlight how incorporating new data impacts in-domain accuracy, cross-domain generalization performance, and thus provide insights into optimal fine-tuning strategies for improving model robustness.

Fine-tuning	**Test**	**Accuracy**		**Macro F1**	
		CS	**SV**	**CS**	**SV**
IPV	IPV	0.859 ± 0.029	0.827 ± 0.000	0.837 ± 0.837	0.793 ± 0.000
	New	0.735 ± 0.005	0.735 ± 0.014	0.444 ± 0.037	0.529 ± 0.026
IPV+1/2New	IPV	0.920 ± 0.026	0.891 ± 0.010	0.911 ± 0.029	0.878 ± 0.008
	New	0.747 ± 0.047	0.747 ± 0.028	0.597 ± 0.098	0.574 ± 0.051

Table 4.6: Effect of incorporating half New dataset into fine-tuning (2 Classes)

Detection As shown in Table 4.6, in the binary classification task, when only IPV data is used for fine-tuning, the model has high accuracy and macro F1 on the IPV test set. However, the generalization ability on the new dataset is poor, with the accuracy dropping to 0.735 (CS and SV), and the macro F1 dropping to 0.444 (CS) and 0.529 (SV).

After introducing half of the new data for fine-tuning, the performance on the IPV dataset is further improved, with the accuracy of CS and SV increasing to 0.920 and 0.891, respectively, and the macro F1 increasing to 0.911 and 0.878, respectively. The generalization ability on the new dataset is also improved, with the accuracy of CS and SV both slightly increasing to 0.747, and the macro F1 improving more significantly, reaching 0.597 (CS) and 0.574 (SV).

Although hybrid fine-tuning improves the performance of the model on IPV and new datasets, the improvement in generalization ability is still limited. The accuracy on the new dataset only increases slightly by +0.012 (CS and SV), but the macro F1 improvement is more obvious (CS +0.153, SV +0.045).

Fine-tuning	**Test**	**Accuracy**		**Macro F1**	
		CS	**SV**	**CS**	**SV**
IPV	IPV	0.637 ± 0.068	0.549 ± 0.045	0.410 ± 0.110	0.175 ± 0.071
	New	0.185 ± 0.042	0.194 ± 0.155	0.078 ± 0.022	0.106 ± 0.074
IPV+1/2New	IPV	0.520 ± 0.017	0.510 ± 0.045	0.210 ± 0.004	0.126 ± 0.036
	New	0.611 ± 0.083	0.509 ± 0.069	0.236 ± 0.073	0.234 ± 0.047

Table 4.7: Effect of incorporating half New dataset into fine-tuning (8 Classes)

Classification Table 4.7 shows the impact of introducing new data during fine-tuning on 8 categories. The results show that fine-tuning only on the IPV dataset, results in poor generalization of the model on new datasets, with CS accuracy as low as 0.185 and SV accuracy as low as 0.194. Especially when there are many categories and the categories are severely imbalanced, fine-tuning only on IPV data is difficult to adapt to unseen data. The macro F1 score is lower (CS: 0.078, SV: 0.106), and the model performs poorly on multi-class classification tasks in new domains compared to the generalization performance of detection tasks (Table 4.4).

To solve this problem, half of the new data is introduced for fine-tuning, and the results show that the generalization ability of the new dataset can be significantly improved. The accuracy on the CS and SV datasets is improved to 0.611 (+0.426) and 0.509 (+0.315), respectively. At the same time, the macro F1 also improves significantly (CS: 0.236, SV: 0.234. However, this improvement comes at the expense of accuracy on the IPV dataset (CS drops by 0.117, SV drops by 0.039). This shows that although mixed-domain data fine-tuning can improve cross-domain performance, it may slightly affect the in-domain performance.

Chapter 5

Discussion

This chapter analyzes the results and reasons for all the experiments we conducted on the Wav2ve2.0 model, and introduces and focuses on multimodal learning, data augmentation, fine-tuning-based generalization, and task-specific transfer strategies for both CS and SV modalities.

We first explore how different fusion strategies affect model performance in Section 5.1, and analyze the early, middle, and late strategies in combination with the two tasks of detection and classification. Section 5.2 studies data augmentation, but unlike the research focus of [10], it mainly emphasizes that the optimal augmentation level varies depending on the dataset and task, and the augmentation strategy should be carefully adjusted according to specific needs. In addition, we discuss the impact of cross-domain fine-tuning on unseen data in Section 5.3. Overall, our research results emphasize the need for customized fusion, augmentation, and training strategies to balance accuracy and generalization, thereby improving the robustness of voice disorder detection and classification models.

5.1 Effectiveness of multimodal learning

Our study has found that multimodal learning, which has great potential power in combining sentence reading modality(CS), and sustained vowel pronunciation(SV) information. By integrating their voice features, multimodal fusion can effectively improve the accuracy, and robustness of detection and classification tasks.

However, the actual effectiveness of multimodal learning, is closely related to specific fusion strategies and task requirements. In the task of voice disorder detection, especially for the mid-level fusion method, it adopts the cross-attention mechanism, to achieve the highest classification accuracy, by learning cross-modal dependency relationships. However, simple early fusion did not significantly outperform with single modal Wav2Vec2.0, indicating if we directly merge data at the original

feature level, it may not fully utilize the complementarity between two modalities. In contrast, the expert of mixture (MoE) method improved the macro-F1 score, indicating its advantage in dealing with class imbalance problems. Another important finding is that SV benefits more from multimodal fusion compared to CS. This may be because the voice acoustic characteristic of SV is relatively limited, and the information content is relatively simple, so it relies more on the additional features provided by CS. However, CS itself contains more contextual information and has relatively low additional requirements for SV. Interestingly, on the new dataset, fusion is not particularly helpful for improving the generalization of unknown data, which may be due to the fitting with the known data.

In voice disorder classification tasks, multimodal fusion does not always bring significant improvements, due to the very small size of the dataset and the unstable distribution of categories. For example, although early fusion improved overall accuracy, the macro-F1 score actually decreased, indicating that class imbalance is still an issue that cannot be ignored. In contrast, MoE's late fusion strategy focuses more on optimizing F1 scores rather than accuracy, indicating that it has certain advantages in balancing class distribution, but may not necessarily improve the overall classification accuracy.

Therefore, relying solely on fusion strategies is not enough to completely solve the generalization problem and category imbalance problem. In order to fully leverage the advantages of multimodal learning in complex classification tasks, additional techniques such as modality-aware weighting, may need to be introduced to more effectively adjust the contribution ratio of each modality, to ensure that the model can achieve more balanced learning between different categories.

5.2 Impact of data augmentation

When designing a data augmentation pipeline, the level of augmentation needs to be careful, and it should match the generalization required for the task.

Unimodal Based on the results in the previous chapter, for the IPV dataset (recorded with the microphone in a standard environment), moderate augmentation (50%) works best for CS, while low augmentation (30%) works better for SV. However, excessive augmentation (70%) may harm performance. This is because adding too much variation introduces a lot of noise, and makes the audio lose its original characteristics, thereby reducing the model's accuracy for similar sounds. This effect is particularly evident in voice classification, where the effect of augmentation is significant. Although the accuracy has improved overall, maintaining class balance remains a challenge for this task.

On the other hand, when evaluating the model generalization on a new dataset

recorded on a mobile device, the situation changes. The higher the augmentation (70% for CS and 50% for SV), the better the results.

Multimodal For the multimodal model combined with data enhancement, this framework can both exploit the complementary information of the two modalities, and improve the robustness of the model, so it achieves the best performance on the IPV dataset. However, the generalization performance on new data is not as good as that of the unimodal augmentation model. This may be because the framework causes overfitting on the IPV dataset, which negatively affects the model's ability to generalize to new datasets.

It can be seen that, although a large strength of augmentation may reduce the accuracy within the same dataset, it helps the model better adapt to new environments, and changes in data distribution. Therefore, augmentation should not be applied in the same way in all datasets or tasks. Instead, it should be tailored to the specific dataset, and specific cases to balance in-domain accuracy and generalization to unseen data.

5.3 Fine-tuning and cross-domain generalization

As expected, fine-tuning on a single dataset (such as IPV) will limit the generalization ability of the model on new datasets. This problem is particularly evident in multi-classification tasks with uneven category distribution. Due to insufficient samples, some categories are difficult for the model to accurately identify, resulting in poor classification results.

To improve the generalization ability, we added some new data during the fine-tuning process. The experimental results show that, this method can indeed significantly improve the generalization ability of the model, especially the improvement of accuracy and macro-F1 score, proving that the introduction of new data helps the model adapt to different data distributions. In addition, this approach has low requirements for the amount of data and is suitable for data shortages in the medical field.

However, the addition of new data has different effects on the performance of the model on the original data (IPV). For the binary classification task, the accuracy of the model on the IPV dataset is improved after adding new data. This may be because the category ratio of the new data is similar to that of the original data, so it does not cause much interference with the learning of the model. In the multi-classification task, the situation is the opposite - the performance on the IPV dataset decreases after adding new data. This may be related to the difference in data distribution. The classification in multi-classification tasks is more complicated. The distribution of new data is quite different from that of

original data, which affects the performance of the model on the original data.

Therefore, when adopting a fine-tuning strategy, it is best to analyze the distribution of the original data first, and then decide how to introduce new data. Reasonable selection of additional data sources, to ensure that they match the original data, can improve generalization capabilities while minimizing the impact on in-domain performance.

Chapter 6

Conclusion

In this study, we explored how multimodal learning, data augmentation, and fine-tuning affect the performance of voice disorder detection and classification using Wav2Vec2.0, an end-to-end Transformer-based model.

We found that combining features from two modalities, CS (sentence reading) and SV (sustained vowels), helped improve the model's performance. However, not all fusion methods were equally effective. Mid-level fusion with cross-attention performed best on the detection task, while simple early fusion worked well on the classification task. The MoE (mixture of experts) method helped balance the class distribution. Interestingly, SV benefited more from multimodal learning than CS, perhaps because SV has fewer features and benefits more from the additional information.

Data augmentation also had a significant impact on the results. We found that moderate augmentation (50%) worked best for CS, while mild augmentation (30%) was more effective for SV and multimodal learning. However, over-augmentation actually degraded performance, probably because it introduced too much noise, making the original audio features harder to distinguish. The effect became more interesting when we tested the model on new data. Stronger augmentation helped the model adapt to unseen data, although it sometimes reduced accuracy on the original dataset. Notably, when combined with data augmentation, unimodal models generalized better to new data than multimodal models. This suggests that multimodal fusion may lead to overfitting of the training dataset. Therefore, augmentation strategies should not be uniformly applied to all tasks. Instead, they should be carefully tuned based on the characteristics of the dataset and the specific task.

We also studied fine-tuning to solve cross-domain problems. For example, in the classification problem, if we only fine-tune on one dataset (such as IPV), the model will have some trouble processing new data. But when we add some new data during the fine-tuning process, the model generalizes better. This provides a

possible solution to the scarcity of medical data. By gradually adding new disease categories to the existing medical model, we can help the model predict new diseases more accurately and reduce the cost of medical research and development. In addition, the multi-class pre-trained model performs better than the model based on binary classification pre-training, which may be because the pre-trained binary model focuses too much on distinguishing "healthy" from "unhealthy", and cannot capture fine-grained features when extended to multi-classification.

Limitation Despite these findings, our work still has limitations:

1. The experiments are based on two modes of homogeneous audio, and the dataset size is still relatively small. The robustness of the model to a wider range of pathological variations requires further validation.

2. Augmentation uses a fixed enhancement ratio (e.g. 30%, 50%, 70%), but in fact adaptive augmentation technology is more suitable for different data types. A more flexible augmentation strategy can improve performance.

3. Although we explored different fusion strategies, our study was limited to the audio-based modality. Multimodal learning coupled with other inputs such as laryngoscopy images could provide deeper insights.

Future work To address these limitations and further improve speech disorder detection and classification models, future research could explore:

1. Larger and more diverse datasets can be explored, including pathological samples that combine sounds with other types of patterns, not limited to audio scenes, such as laryngoscope images, to verify the robustness and adaptability of our model.

2. Adaptive augmentation methods such as AutoAugment or RandAugment can be explored to automatically adjust the augmentation strength, to dynamically adapt it to the data distribution.

3. The fusion strategy and augmentation framework can also be used for multi-classification tasks, such as identifying different pathological conditions. It can also be applied to other pre-trained models, like HuBERT and WavLM. This helps to better evaluate how effective and useful the model is.

Ethical considerations As AI models become more common in medical applications, we need to carefully consider their ethical implications. Voice data is highly sensitive, and issues of privacy, security, and potential abuse cannot be ignored. A major concern is that attackers can extract private information from model

representations through inversion attacks. To prevent this, privacy-preserving techniques such as differential privacy and secure federated learning should be explored. When publishing and deploying these models, we must also ensure that they are fair and unbiased, as failure to generalize across different patient populations, can lead to misdiagnosis and unequal medical outcomes. Most importantly, AI-powered diagnostic tools should be used as assistive tools, not replacements for human medical experts. Automation can also carry the risk of bias, and over-reliance on AI predictions can make it difficult to critically evaluate, making transparency and explainability essential—models should be designed in a way, that users can understand and trust their decisions.

In the future, research should focus not only on improving accuracy, but also on ensuring fairness, accountability, and ethical responsibility. By addressing these challenges, we can create AI-driven tools that truly support clinicians and provide better and safer care for patients.

Bibliography

[1] Nelson Roy, Ray M Merrill, Steven D Gray, and Elaine M Smith. «Voice disorders in the general population: prevalence, risk factors, and occupational impact». In: *The Laryngoscope* 115.11 (2005), pp. 1988–1995.

[2] Neil Bhattacharyya. «The prevalence of voice problems among adults in the United States». In: *The Laryngoscope* 124.10 (2014), pp. 2359–2362.

[3] Nelson Roy and Diane M Bless. «Personality traits and psychological factors in voice pathology: a foundation for future research». In: *Journal of Speech, Language, and Hearing Research* 43.3 (2000), pp. 737–748.

[4] Seth M Cohen, Jaewhan Kim, Nelson Roy, Carl Asche, and Mark Courey. «The impact of laryngeal disorders on work-related dysfunction». In: *The Laryngoscope* 122.7 (2012), pp. 1589–1594.

[5] Jeffrey L Cutler and Thomas Cleveland. «The clinical usefulness of laryngeal videostroboscopy and the role of high-speed cinematography in laryngeal evaluation». In: *Current Opinion in Otolaryngology & Head and Neck Surgery* 10.6 (2002), pp. 462–466.

[6] Pedram Daraei, Craig R. Villari, Adam D. Rubin, Alexander T. Hillel, Edie R. Hapner, Adam M. Klein, and III Johns Michael M. «The Role of Laryngoscopy in the Diagnosis of Spasmodic Dysphonia». In: *JAMA Otolaryngology–Head & Neck Surgery* 140.3 (Mar. 2014), pp. 228–232. URL: https://doi.org/10.1001/jamaoto.2013.6450.

[7] Mathieu Bergeron, Robert J. Fleck, Stephanie R. C. Zacharias, Meredith E. Tabangin, and Alessandro de Alarcon. «The Value of Dynamic Voice CT Scan for Complex Airway Patients Undergoing Voice Surgery». In: *Annals of Otology, Rhinology & Laryngology* 128.10 (2019), pp. 885–893. DOI: 10.1177/0003489419846138.

[8] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. «AI in health and medicine». In: *Nature medicine* 28.1 (2022), pp. 31–38.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[10] Alkis Koudounas, Gabriele Ciravegna, Marco Fantini, Erika Crosetti, Giovanni Succo, Tania Cerquitelli, and Elena Baralis. «Voice Disorder Analysis: a Transformer-based Approach». In: *Interspeech 2024*. interspeech_2024. ISCA, Sept. 2024, pp. 3040–3044. DOI: 10.21437/interspeech.2024-1122. URL: http://dx.doi.org/10.21437/Interspeech.2024-1122.

[11] Christopher L Payten, Greg Chiapello, Kelly A Weir, and Catherine J Madill. «Frameworks, terminology and definitions used for the classification of voice disorders: a scoping review». In: *Journal of Voice* (2022).

[12] Katherine Verdolini, Clark A Rosen, and Ryan C Branski. *Classification manual for voice disorders-I*. Psychology Press, 2014.

[13] Sevtap Akbulut et al. «Basics of Voice Disorders». In: *Phoniatrics I: Fundamentals – Voice Disorders – Disorders of Language and Hearing Development*. Ed. by Antoinette am Zehnhoff-Dinnesen, Bozena Wiskirska-Woznica, Katrin Neumann, and Tadeus Nawka. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 193–238. DOI: 10.1007/978-3-662-46780-0_4.

[14] Eman Ezzat, Hossam El-Dessouky, Marwa GA El-Hameed, and Mohamed Baraka. «Role of functional MRI in assessment of voice, language, and speech disorders». In: *Menoufia Medical Journal* 32.3 (2019), pp. 763–769.

[15] Laura Verde, Giuseppe De Pietro, and Giovanna Sannino. «Voice Disorder Identification by Using Machine Learning Techniques». In: *IEEE Access* 6 (2018), pp. 16246–16255. DOI: 10.1109/ACCESS.2018.2816338.

[16] Lady Catherine Cantor-Cutiva, Sai Aishwarya Ramani, Patrick R. Walden, and Eric J. Hunter. «Screening of Voice Pathologies: Identifying the Predictive Value of Voice Acoustic Parameters for Common Voice Pathologies». In: *Journal of Voice* (2023). ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2023.12.005.

[17] Leonardo Wanderley Lopes, Layssa Batista Simões, Jocélio Delfino da Silva, Deyverson da Silva Evangelista, Ana Celiane da Nóbrega e Ugulino, Priscila Oliveira Costa Silva, and Vinícius Jefferson Dias Vieira. «Accuracy of Acoustic Analysis Measurements in the Evaluation of Patients With Different Laryngeal Diagnoses». In: *Journal of Voice* 31.3 (2017), 382.e15–382.e26. ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2016.08.015.

[18] Shiva Ebrahimian Dehaghani, Mahmood Bijankhan, Fateme Arjmandpur, Hoda Mozoonei, Omid Yaghini, and Mohadeseh Ebrahimian. «Description of Three Time-Domain Speech Features in Children with Down syndrome: A pilot study». In: *Journal of Rehabilitation Sciences & Research* 7.2 (2020), pp. 75–79. URL: https://jrsr.sums.ac.ir/article_46597.html.

[19] Jung-Won Lee, Hong-Goo Kang, Jeung-Yoon Choi, and Young-Ik Son. «An investigation of vocal tract characteristics for acoustic discrimination of pathological voices». In: *BioMed research international* 2013.1 (2013), p. 758731.

[20] Ghulam Muhammad, Mansour Alsulaiman, Awais Mahmood, and Zulfiqar Ali. «Automatic voice disorder classification using vowel formants». In: *2011 IEEE international conference on multimedia and expo*. IEEE. 2011, pp. 1–6.

[21] Victoria S McKenna and Cara E Stepp. «The relationship between acoustical and perceptual measures of vocal effort». In: *The Journal of the Acoustical Society of America* 144.3 (2018), pp. 1643–1658.

[22] Duy Duong Nguyen, Daniel Novakovic, and Catherine Madill. «Voice disorder discrimination using vowel acoustic measures in female speakers». In: *International Journal of Language & Communication Disorders* (2024).

[23] Lindasalwa Muda. «Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques». In: *arXiv preprint arXiv:1003.4083* (2010).

[24] Mahadevaswamy Shanthamallappa, Kiran Puttegowda, Naveen Kumar Hullahalli Nannappa, and Sudheesh Kannur Vasudeva Rao. «Robust Automatic Speech Recognition Using Wavelet-Based Adaptive Wavelet Thresholding: A Review». In: *SN Computer Science* 5.2 (2024), p. 248.

[25] Zulfiqar Ali, Mansour Alsulaiman, Ghulam Muhammad, Irraivan Elamvazuthi, and Tamer A. Mesallam. «Vocal fold disorder detection based on continuous speech by using MFCC and GMM». In: *2013 7th IEEE GCC Conference and Exhibition (GCC)*. 2013, pp. 292–297. DOI: 10.1109/IEEEGCC.2013.6705792.

[26] Xiaoping Xie, Hao Cai, Can Li, Yu Wu, and Fei Ding. «A Voice Disease Detection Method Based on MFCCs and Shallow CNN». In: *Journal of Voice* (2023). ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2023.09.024.

[27] Fethi Amara, Mohamed Fezari, and Bourouba Hocine. «An Improved GMM-SVM System based on Distance Metric for Voice Pathology Detection». In: *Applied Mathematics & Information Sciences* 10 (May 2016), pp. 1061–1070. DOI: 10.18576/amis/100324.

[28] Raphael Torres Santos Carvalho, Charles Casimiro Cavalcante, and Paulo César Cortez. «Wavelet transform and artificial neural networks applied to voice disorders identification». In: *2011 Third World Congress on Nature and Biologically Inspired Computing*. 2011, pp. 371–376. DOI: 10.1109/NaBIC.2011.6089256.

[29] L. Salhi, M. Talbi, and A. Cherif. «Voice Disorders Identification Using Hybrid Approach: Wavelet Analysis and Multilayer Neural Networks». In: *International Journal of Electrical and Computer Engineering* 2.9 (2008), pp. 3003–3012. ISSN: eISSN: 1307-6892. URL: https://publications.waset.org/vol/21.

[30] Jinxiang Liu, Tiejun Wang, Andrew Skidmore, Yaqin Sun, Peng Jia, and Kefei Zhang. «Integrated 1D, 2D, and 3D CNNs Enable Robust and Efficient Land Cover Classification from Hyperspectral Imagery». In: *Remote Sensing* 15.19 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15194797. URL: https://www.mdpi.com/2072-4292/15/19/4797.

[31] Irfan Aziz. «Deep learning: an overview of Convolutional Neural Network (CNN)». In: (2020).

[32] Yang Liusong and Du Hui. «Voice quality evaluation of singing art based on 1DCNN model». In: *Mathematical Problems in Engineering* 2022.1 (2022), p. 2074844.

[33] NURUL NADHRAH KAMARUZAMAN et al. «SMOTE-2DCNN FOR ENHANCING SPEECH EMOTION RECOGNITION». In: *Journal of Theoretical and Applied Information Technology* 102.13 (2024).

[34] ZhangFang Hu, XingTong Si, Yuan Luo, ShanShan Tang, and Fang Jian. «Speaker Recognition Based on 3DCNN-LSTM.» In: *Engineering Letters* 29.2 (2021).

[35] Mazin Abed Mohammed, Karrar Hameed Abdulkareem, Salama A. Mostafa, Mohd Khanapi Abd Ghani, Mashael S. Maashi, Begonya Garcia-Zapirain, Ibon Oleagordia, Hosam Alhakami, and Fahad Taha AL-Dhief. «Voice Pathology Detection and Classification Using Convolutional Neural Network Model». In: *Applied Sciences* 10.11 (2020). ISSN: 2076-3417. DOI: 10.3390/app10113723.

[36] Rumana Islam, Esam Abdel-Raheem, and Mohammed Tarique. «Voice pathology detection using convolutional neural networks with electroglottographic (EGG) and speech signals». In: *Computer Methods and Programs in Biomedicine Update* 2 (2022), p. 100074. ISSN: 2666-9900. DOI: https://doi.org/10.1016/j.cmpbup.2022.100074. URL: https://www.sciencedirect.com/science/article/pii/S2666990022000258.

[37] Umesh Kumar Lilhore et al. «Hybrid CNN-LSTM model with efficient hyperparameter tuning for prediction of Parkinson's disease». In: *Scientific Reports* 13.1 (2023), p. 14605.

[38] Rimah Amami, Rim Amami, Chiraz Trabelsi, Sherin Hassan Mabrouk, and Hassan A Khalil. «A robust voice pathology detection system based on the combined bilstm–cnn architecture». In: *MENDEL*. Vol. 29. 2. 2023, pp. 202–210.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[40] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. «wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations». In: *CoRR* abs/2006.11477 (2020). arXiv: 2006.11477. URL: https://arxiv.org/abs/2006.11477.

[41] Yuan Gong, Yu-An Chung, and James R. Glass. «AST: Audio Spectrogram Transformer». In: *CoRR* abs/2104.01778 (2021). arXiv: 2104.01778. URL: https://arxiv.org/abs/2104.01778.

[42] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. «HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units». In: *CoRR* abs/2106.07447 (2021). arXiv: 2106.07447. URL: https://arxiv.org/abs/2106.07447.

[43] Sanyuan Chen et al. «WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing». In: *CoRR* abs/2110.13900 (2021). arXiv: 2110.13900. URL: https://arxiv.org/abs/2110.13900.

[44] Farhad Javanmardi, Sudarsana Reddy Kadiri, and Paavo Alku. «A comparison of data augmentation methods in voice pathology detection». In: *Computer Speech & Language* 83 (2024), p. 101552. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2023.101552. URL: https://www.sciencedirect.com/science/article/pii/S0885230823000712.

[45] Olusola O. Abayomi-Alli, Robertas Damaševičius, Atika Qazi, Mariam Adedoyin-Olowe, and Sanjay Misra. «Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review». In: *Electronics* 11.22 (2022). ISSN: 2079-9292. URL: https://www.mdpi.com/2079-9292/11/22/3795.

[46] Máté Hireš, Matej Gazda, Lukáš Vavrek, and Peter Drotár. «Voice-specific augmentations for Parkinson's disease detection using deep convolutional neural network». In: *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE. 2022, pp. 000213–000218.

[47] Dustin Boswell. «Introduction to support vector machines». In: *Departement of Computer Science and Engineering University of California San Diego* 11 (2002), pp. 16–17.

[48] Guodong Guo and S.Z. Li. «Content-based audio classification and retrieval by support vector machines». In: *IEEE Transactions on Neural Networks* 14.1 (2003), pp. 209–215. DOI: 10.1109/TNN.2002.806626.

[49] Lie Lu, Stan Z. Li, and Hong-Jiang Zhang. «Content-based audio segmentation using support vector machines». In: *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.* 2001, pp. 749–752. DOI: 10.1109/ICME.2001.1237830.

[50] Pak Ho Leung, Kwok Tai Chui, Kenneth Lo, and Patricia Ordóñez de Pablos. «Chapter 13 - A support vector machine–based voice disorders detection using human voice signal». In: *Artificial Intelligence and Big Data Analytics for Smart Healthcare*. Ed. by Miltiadis D. Lytras, Akila Sarirete, Anna Visvizi, and Kwok Tai Chui. Next Gen Tech Driven Personalized Med&Smart Healthcare. Academic Press, 2021, pp. 197–208. ISBN: 978-0-12-822060-3. DOI: https://doi.org/10.1016/B978-0-12-822060-3.00014-0.

[51] Roozbeh Behroozmand and Farshad Almasganj. «Optimal selection of wavelet-packet-based features using genetic algorithm in pathological assessment of patients' speech signal with unilateral vocal fold paralysis». In: *Computers in Biology and Medicine* 37.4 (2007). Wavelet-based Algorithms for Medical Problems, pp. 474–485. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2006.08.016.

[52] T. Cover and P. Hart. «Nearest neighbor pattern classification». In: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

[53] R. Arefi Shirvan and E. Tahami. «Voice analysis for detecting Parkinson's disease using genetic algorithm and KNN classification method». In: *2011 18th Iranian Conference of Biomedical Engineering (ICBME)*. 2011, pp. 278–283. DOI: 10.1109/ICBME.2011.6168572.

[54] Lili Chen, Chaoyu Wang, Junjiang Chen, Zejun Xiang, and Xue Hu. «Voice Disorder Identification by using Hilbert-Huang Transform (HHT) and K Nearest Neighbor (KNN)». In: *Journal of Voice* 35.6 (2021), 932.e1–932.e11. ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2020.03.009.

[55] Lawrence R Rabiner. «A tutorial on hidden Markov models and selected applications in speech recognition». In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[56] A.A. Dibazar, S. Narayanan, and T.W. Berger. «Feature analysis for automatic detection of pathological speech». In: *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] [Engineering in Medicine and Biology*. Vol. 1. 2002, 182–183 vol.1. DOI: 10.1109/IEMBS.2002.1134447.

[57] Julián D. Arias-Londoño, Juan I. Godino-Llorente, Nicolás Sáenz-Lechón, Víctor Osma-Ruiz, and Germán Castellanos-Domínguez. «An improved method for voice pathology detection by means of a HMM-based feature space transformation». In: *Pattern Recognition* 43.9 (2010), pp. 3100–3112. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2010.03.019.

[58] Douglas A Reynolds and Richard C Rose. «Robust text-independent speaker identification using Gaussian mixture speaker models». In: *IEEE transactions on speech and audio processing* 3.1 (1995), pp. 72–83.

[59] Zulfiqar Ali, Ghulam Muhammad, and Mohammed F. Alhamid. «An Automatic Health Monitoring System for Patients Suffering From Voice Complications in Smart Cities». In: *IEEE Access* 5 (2017), pp. 3900–3908. DOI: 10.1109/ACCESS.2017.2680467.

[60] Maillikarjun S Holi et al. «Wavelet transform features to hybrid classifier for detection of neurological-disordered voices». In: *Journal of Clinical Engineering* 42.2 (2017), pp. 89–98.

[61] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG]. URL: https://arxiv.org/abs/1912.05911.

[62] Jerry Joy, Aparna Kannan, Shreya Ram, and S Rama. «Speech emotion recognition using neural network and MLP classifier». In: *Ijesc* 2020 (2020), pp. 25170–25172.

[63] Arjun Ghosh, Nanda Dulal Jana, Saurav Mallik, and Zhongming Zhao. «Designing optimal convolutional neural network architecture using differential evolution algorithm». In: *Patterns* 3.9 (2022), p. 100567. ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2022.100567. URL: https://www.sciencedirect.com/science/article/pii/S2666389922001787.

[64] Tianqi Chen and Carlos Guestrin. «XGBoost: A Scalable Tree Boosting System». In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Vol. 11. KDD '16. ACM, Aug. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[65] Juan Rafael Orozco, Julian D. Arias-Londoño, J. Vargas-Bonilla, María González-Rátiva, and Elmar Noeth. «New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease». In: May 2014.

[66] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. «wav2vec: Unsupervised Pre-training for Speech Recognition». In: *CoRR* abs/1904.05862 (2019). arXiv: 1904.05862. URL: http://arxiv.org/abs/1904.05862.

[67] Alexei Baevski, Steffen Schneider, and Michael Auli. «vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations». In: *CoRR* abs/1910.05453 (2019). arXiv: 1910.05453. URL: http://arxiv.org/abs/1910.05453.

[68] Jialu Li, Vimal Manohar, Pooja Chitkara, Andros Tjandra, Michael Picheny, Frank Zhang, Xiaohui Zhang, and Yatharth Saraf. *Accent-Robust Automatic Speech Recognition Using Supervised and Unsupervised Wav2vec Embeddings*. 2021. arXiv: 2110.03520 [eess.AS]. URL: https://arxiv.org/abs/2110.03520.

[69] Eric Jang, Shixiang Gu, and Ben Poole. *Categorical Reparameterization with Gumbel-Softmax*. 2017. arXiv: 1611.01144 [stat.ML]. URL: https://arxiv.org/abs/1611.01144.

[70] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. «Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks». In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 369–376.

[71] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. «ImageNet: A large-scale hierarchical image database». In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[72] Yuan Gong, Yu-An Chung, and James Glass. *PSLA: Improving Audio Event Classification with Pretraining, Sampling, Labeling, and Aggregation*. Feb. 2021. DOI: 10.48550/arXiv.2102.01243.

[73] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. «Multimodal Machine Learning: A Survey and Taxonomy». In: *CoRR* abs/1705.09406 (2017). arXiv: 1705.09406. URL: http://arxiv.org/abs/1705.09406.

[74] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. «Multimodal deep learning for biomedical data fusion: a review». In: *Briefings in Bioinformatics* 23.2 (2022), bbab569.

[75] Hossein Jadvar and Patrick M Colletti. «Competitive advantage of PET/MRI». In: *European journal of radiology* 83.1 (2014), pp. 84–94.

[76] Kim-Han Thung, Pew-Thian Yap, and Dinggang Shen. «Multi-stage diagnosis of Alzheimer's disease with incomplete multimodal data via multi-task deep learning». In: *International Workshop on Deep Learning in Medical Image Analysis*. Springer. 2017, pp. 160–168.

[77] Loukas Ilias, Dimitris Askounis, and John Psarras. «Detecting dementia from speech and transcripts using transformers». In: *Computer Speech & Language* 79 (2023), p. 101485. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2023.101485.

[78] John Arevalo, Thamar Solorio, Manuel Montes-y-Gómez, and Fabio A. González. *Gated Multimodal Units for Information Fusion*. 2017. arXiv: 1702.01992 [stat.ML]. URL: https://arxiv.org/abs/1702.01992.

[79] Gail B Kempster, Bruce R Gerratt, Katherine Verdolini Abbott, Julie Barkmeier-Kraemer, and Robert E Hillman. «Consensus auditory-perceptual evaluation of voice: development of a standardized clinical protocol». In: (2009).